ЗАТВЕРДЖЕНО
Наказ Вищого навчального закладу Укоопспілки «Полтавський університет економіки і торгівлі»
08 липня 2015 року № 152-Н

Форма № П-4.04

ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД УКООПСПІЛКИ
«ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ»

Інститут економіки, управління та інформаційних технологій

Форма навчання денна

Кафедра економічної кібернетики, бізнес економіки та інформаційних систем

Завідувач кафедри д.е.н., проф.
_____ М.Є.Рогоза
(підпис, ініціали та прізвище)
«_____»_____2019 р.

ДИПЛОМНА РОБОТА

на тему:
«Автоматизація процесів верифікації бізнес критичних програмних систем управління економічним обєктом»

зі спеціальності 051 „Економіка”
освітня програма «Економічна кібернетика»
(шифр та назва)

Виконавець роботи Вівтоніченко Ярослав Вячеславович
(прізвище, ім'я, по батькові)
_____
(підпис, дата)

Науковий керівник проф., д.е. н. Рогоза Микола Єгорович
(науковий ступінь, вчене звання, прізвище, ім'я, по батькові)
_____
(підпис, дата)

ПОЛТАВА 2019

Перелік позначень та скорочень

AD	Architecture Description
ASIC	Application-specific integrated circuit
CPLD	Complex Programmable Logic Device
DD	Detailed Description
ED	Electronic Design
MC/DC	Modified condition\decision coverage
FBL	Function Block Library
FEC	Focused Exprassion Coverage
FPGA	Field Programmable Gate Array
FT	Functional Testing
FB	Function Block
OVVP	Overall Verification and Validation Plan
VHDL	Very high speed integrated circuit Hardware Description Language
VST	Verification Support Tool
RTL	Registry Transfer Level
ЕСПД	Едина система програмної документації
ІЗ	Інструментальний засіб
ІУС	Інформаційно-керуюча система
ПТК	Программмно-технічний комплекс
ТО	Тестуючий об’єкт
SCA	Static Code Analysis
CR	Code Review

ВСТУП

В даний час створення високоякісного програмного забезпечення є однією з найважливіших завдань розвитку науки та виробництва. Від того, наскільки вдало зроблено програмне забезпечення системи, залежить в кінцевому результаті її життєздатність. Проте, в зв’язку з тим, що тільки деякі суттєві властивості програмного забезпечення можуть бути виміряні безпосередньо і оцінені кількісними показниками, забезпечення належного рівня його якості набуває першочергового значення.

Таким чином, однією з актуальних та важливих задач програмної інженерії є реалізація ефективної технології забезпечення необхідного рівня якості програмних систем. Вирішення цієї задачі є особливо важливим для бізнес критичних програмних систем [ ].

Об’єкт дослідження – процеси верифікації спеціалізованого програмного забезпечення об'єктів контролю і управління енергоблоків українських АЕС.

Предмет дослідження – методики та процедури реалізації процесів верифікації програмного забезпечення на ТОВ «НВП Радікс» та напрямки їх вдосконалення.

Мета дипломної роботи: підвищення економічної ефективності застосування методів і засобів верифікації спеціалізованого програмного забезпечення ТОВ «НВП Радікс» на основі їх комплексної автоматизації.

Для досягнення мети дипломної роботи поставлено такі **завдання** :

- визначити базові поняття верифікації та валідації програмного забезпечення, а саме: основні методології верифікації та валідації, технології створення testbench та testcase;

- розглянути питання організації процесів верифікації бізнес-критичних програмних систем управління економічним об’єктом, моделі та методичні підходи аналізу та управління ризиками у життєвому циклі верифікації бізнес критичних систем управління та їх програмного забезпечення, планування та ефективність

використання автоматизації процесів верифікації бізнес критичних систем управління економічним об’єктом;

- дослідити діяльність та стан бізнес критичних програмних систем управління економічним об’єктом, таким, як атомна електростанція;

- провести аналіз стану діяльності та актуальність бізнес критичних програмних систем управління економічним об’єктом;

- виконати оцінку ефективності дії процесів життєвого циклу бізнес критичних програмних систем управління економічним об’єктом;

- розглянути приклади моделей та методів життєвого циклу верифікації бізнес критичних програмних систем управління економічним об’єктом, методичні підходи аналізу та управління ризиками у життєвому циклі верифікації бізнес критичних систем управління та їх програмного забезпечення;

- запропонувати можливі напрямки підвищення економічної ефективності застосування методів і засобів верифікації спеціалізованого програмного забезпечення ТОВ «НВП Радікс» на основі їх комплексної автоматизації.

Тематика дипломної роботи набуває особливої актуальності в останнє десятиліття, що визначається значної увагою до неї в науковій спільноті. Варто зазначити доробки таких провідних науковців в визначеній сфері, як: Ястребенецький М.А., Харченко В.С., Скляр В.В., Вовк О.Б., Георгіаді Н.Г., Гушко С.В., Шайкан А.В.

Основним орієнтиром при створенні критичних програмних систем є стандарт IEC 61508 -- Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems.

Дипломна робота складається з: вступу, трьох розділів, що включають 9 підрозділів, висновків, списку використаних джерел із 56 найменувань.

У тексті дипломної роботи міститься 22 таблиці і 20 рисунків. Загальний обсяг роботи 167 аркушів.

Розділ 1. ОРГАНІЗАЦІЯ ПРОЦЕСІВ ВЕРИФІКАЦІЇ БІЗНЕС КРИТИЧНИХ ПРОГРАМНИХ СИСТЕМ УПРАВЛІННЯ ЕКОНОМІЧНИМ ОБ’ЄКТОМ

Розділ 1. Процеси верифікації критичних програмних систем як основа управління економічним об’єктом

Під верифікацією програмного забезпечення (ПЗ) розуміють процес надання об'єктивних доказів того, що програмне забезпечення та його асоційовані продукти відповідають вимогам (наприклад, щодо правильності, повноти, послідовності та точність) для всіх заходів життєвого циклу протягом кожного життєвого циклу (придбання, постачання, розвиток, експлуатація та обслуговування), відповідають стандартам, практиці та умовам під час процесів життєвого циклу та успішно завершити кожну діяльність життєвого циклу та задовольнити всі критерії для початку успішного життєвого циклу дії (наприклад, правильне складання програмного забезпечення).

При цьому мається на увазі не тільки тестування самої програми, але й аудит проекту, користувацької і технічної документації тощо .

Відповідно до стандарту ISO 9000 термін верифікація означає"підтвердження на основі подання об'єктивних свідчень того, що встановлені вимоги були виконані".

Тестування – це перевірка роботи програм з даними, подібними реальним, котрі будуть оброблятися в процесі експлуатації системи. Несправності в роботі ПЗ виявляються при аналізі вихідних даних, серед яких виділяються і досліджуються аномальні.

В даній роботі в якості економічного обєкта розглядається АЕС, а субєктом вуступає процес верифікації програмного забеспечення безпечного функціонування АЕС. Дане питання є дуже важливим як в плані безпечної роботи АЕС, так і в плані економічному: Один день простою АЕС обходиться в 1 мільйон гривень.

Удосконалення процесів верифікації дає можливість не лише скоротити час провірки компонентів системи безпеки АЕС з 50 люд./год. на компонент до 20

люд./год., а й підвищити якість такої перевірки, що є дуже важливим в плані безпеки життя.

Висновки

Охарактеризовано питання верифікації бізнес критичних програмних систем управління економічним об’єктом. В результаті аналізу основних процесів верифікації, можна зробити наступні висновки:

- необхідними умовами досягнення високого рівня надійності та безпеки критичного ПЗ є наявність адекватного (ефективного) нормативно-методичного забезпечення та широкомасштабне застосування інструментальних засобів підтримки процесів кваліфікаційних випробувань (експертизи), що відображають сучасний динамічний розвиток стандартизації у сфері інформаційних технологій та програмної інженерії;
- надійність та безпека інформаційно-керуючих систем критичного призначення суттєво залежать від якості програмного забезпечення (ПЗ), за допомогою якого виконуються критичні функції. Приховані дефекти (дефекти, що не були виявлені при тестуванні та верифікації) критичного ПЗ являються факторами ризику відмови системи;
- незалежна верифікація критичного ПЗ, що підтверджує виконання заявлених функцій та дає оцінку вірогідності наявності прихованих дефектів, є необхідною умовою нормативних вимог для різних галузей;
- удосконалення процесів верифікації дає можливість не лише скоротити час перевірки компонентів системи безпеки АЕС з 50 люд./год. на компонент до 20 люд./год., а й підвищити якість такої перевірки, що є дуже важливим в плані безпеки життя.

1.2. Концептуальні підходи організації процесів верифікації бізнес-критичних програмних систем управління економічним об’єктом на основі вимог до їх життєвого циклу

Надійність та безпека інформаційно керуючих систем критичного призначення суттєво залежать від якості програмного забезпечення (ПЗ), за допомогою якого виконуються критичні функції. Приховані дефекти (дефекти, що не були виявлені при тестуванні та верифікації) критичного ПЗ являються факторами ризику відмови системи. Незалежна верифікація критичного ПЗ, що підтверджує виконання заявлених функцій та дає оцінку вірогідності наявності прихованих дефектів, є необхідною умовою нормативних вимог для різних галузей.

З цієї точки зору основними проблемами є: надійність незалежної верифікації,

оцінювання вірогідності прихованих дефектів, повнота тестового покриття

для критичного ПЗ та, як результат,кількісна оцінка функціональної безпеки.

Незалежна верифікація та валідація є ключовою методикою кваліфікаційних випробувань критичного ПЗ. Її проведення є обов’язковою нормативною вимогою в сферах критичної діяльності, таких як атомна енергетика ("Software for computer based systems important to safety in nuclear power plants"- Серія стандартів МАГАТЕ з безпеки), космічна галузь (стандарти ECSS-Q-40B, ECSS-Q-80B) та інші.

Незалежна верифікація в обсязі кваліфікаційних випробувань ПЗ ІКС вирішальним чином визначає реальні можливості в забезпеченні необхідного рівня безпеки і якості ІКС критичного застосування в цілому.

Основу роботи ISO складають процедури, які відомі як ISO/IEC. Переклад з англійської терміну "верифікація" дає певне його тлумачення: verification – перевірка [6]. Щоб було простіше зрозуміти, можна навести приклад типової верифікації: тестування програми або проведення випробування обладнання. Відповідно до певних вимог, що висуваються до об’єкта верифікації проводять випробування і фіксують, чи дотримані вимоги. Результат верифікації – це відповідь на питання "Чи відповідає об’єкт вимогам?".

Етапи верифікації.

1 етап. Аналіз об'єкта верифікації.

1.1. Визначення категорії безпеки функцій виконуваних об’єктом верифікації.

1.2. Визначення характеристик об`єкта верифікації, що впливають на обсяг верифікації.

2 етап. Проведення верифікації

2.1. Визначення ступеня незалежності,кваліфікації та відповідальності осіб, які проводять верифікацію

2.2. Визначення, розробка та затвердження процедур проведення верифікації з визначенням критеріїв для аналізу, застосуванням відповідного устаткування та необхідних документів:технічних умов, програм і методик випробувань, інструкцій тощо

2.3. Визначення заходів по усуненню недоліків, виявлених у процесі верифікації

2.4. Визначення заходів щодо документування процесу верифікації

3 етап. Проведення верифікації.

3.1. Виконання завдань верифікації: проведення необхідних спостережень, контролю і вимірювання, випробування тощо із застосуванням конкретних методів і процедур. При цьому умови, в яких проводяться випробування, можуть бути реальними або змодельованими (наприклад, перевірка з використанням різних лабораторних стендів) в лабораторних умовах для імітації реальних умов експлуатації.

3.2. Усунення недоліків, виявлених в процесі верифікації.

3.3. Документування виконаних дій, включаючи дії з усунення недоліків: оформлення та подання в необхідному вигляді результатів спостереження, контролю, вимірювань, випробувань і т. п. (тобто діяльності, яка здійснюється

для встановлення придатності, адекватності, результативності даної діяльності (об'єкта) для досягнення встановлених цілей). Як правило, це протоколи приймально-здавальних випробувань, акти періодичних, типових випробувань, випробувань на надійність, акти огляду робіт і приймання конструкцій, виконавча документація тощо.

3.4. Висновок за результатами верифікації про відповідність програмних систем висунутим вимогам

4 етап. Висновок за результатами верифікації про відповідність об’єкта верифікації висунутим вимогам.

4.1. Аналіз результатів: аналіз отриманих даних, оцінка їх об'єктивності, повноти і достатності для прийняття рішення про об’єкт верифікації до застосування та її використання за призначенням

4.2. Прийняття рішення про придатність об’єкта верифікації до застосування та використання за призначенням та оформлення документів, що засвідчують приймання продукції.

Верифікація дозволяє своєчасно провести коригувальні та попереджувальні

дії для усунення невідповідностей, що були виявленні, і відповідно уникнути або звести до мінімуму претензії зовнішніх та внутрішніх споживачів, покращити умови експлуатації та використання об’єкта верифікації.

Приховані дефекти критичного ПЗ є суттєвим фактором ризику аварійних ситуацій для системи в цілому. Проведення незалежної верифікації критичного ПЗ - обов'язкова вимога міжнародної нормативної бази.

Метою верифікації є досягнення гарантії того, що верифікується об'єкт (вимога або програмний код), реалізований без непередбачених функцій.

Існують дві основні методики перевірки та аналізу систем в процесах верифікації та атестації: інспектування і автоматичний аналіз - це статичні методи, які можуть виконуватися на всіх етапах процесу розробки системи, а так само тестування - це динамічний метод, який виконується якщо вже створена виконується програма, тобто на етапі реалізації системи і після завершення її реалізації.

Тестування (testing)-це процес виконання програми (або частини програми) з наміром (або метою) знайти помилки. Налагодження (debugging) не є різновидом тестування. Хоча слова «налагодження» і «тестування» часто використовуються як синоніми, під ними маються на увазі різні види діяльності. Тестування - діяльність, спрямована на виявлення помилок; налагодження спрямована на встановлення

точної природи відомої помилки, а потім - на виправлення цієї помилки. Ці два види діяльності пов'язані - результати тестування є вихідними даними для налагодження. Налагодження - складний процес і це обумовлено наступними причинами: від програміста потрібні глибокі знання специфіки управління використовуваними технічними засобами, операційної системи, середовища та мови програмування, реалізованих процесів, природи і специфіки різних помилок, методик налагодження і відповідних програмних засобів. Так само складність відладки може бути обумовлена взаємовпливом помилок у різних частинах програми. Знайдена програмістом помилка виправляється, після чого проводиться повторне тестування, так як в процесі виправлення не виключається можливість появи нових помилок. Повне повторне тестування займає багато часу і як наслідок є дорогим, тому система розбивається на окремі частини і повторно тестується тільки та частина, а так само пов'язані з нею інші частини, де виявлена помилка.

Верифікація та атестація систем, критичних щодо забезпечення безпеки, має багато спільного з тестуванням будь-яких систем з високими вимогами надійності. Щоб виявити найбільшу кількість помилок, слід застосовувати всебічне тестування, а при оцінці безпеки використовувати статичні методи тестування. Однак внаслідок надзвичайно низької частоти відмов, властивих багатьом КС, за допомогою статичного тестування не завжди вдається кількісно оцінити безвідмовність, так як для цього потрібно дуже велике число тестів. Ці тести лише дають підставу вважати ту чи іншу КС безпечною.

Усі докази безпеки системи будуються на наступному припущенні: кількість помилок у системі, які призводять до аварійних ситуацій, набагато менше загального числа помилок в системі. Забезпечення безпеки має зосередитися на виявленні потенційно небезпечних помилок. Якщо виявляється, що ці помилки не виявляються або виявляються, але не призводять до серйозних наслідків, то система вважається надійною. Докази правильності програм були запропоновані в якості методів верифікації ПЗ більше 25 років тому. Однак ці методи в основному використовуються тільки в лабораторіях. Практичні проблеми побудови докази правильності ПЗ настільки складні, що деякі організації вважають використання

даних методів у процесі розробки звичайних систем невиправдано дорогим. Але, як зазначалося раніше, для ряду КС економічно вигідно використовувати докази правильності системи, ніж ліквідувати наслідки відмов. Незважаючи на те, що для більшості систем розробляти докази правильності нерентабельно, іноді виникає необхідність розробити докази безпеки, що демонструють відповідність даної програми вимогам щодо забезпечення безпеки. При доказі безпеки необов'язково доводити відповідність програми специфікації. Необхідно тільки показати, що виконання програми не призводить до збоїв з небезпечними наслідками.

Необхідними умовами досягнення високого рівня надійності та безпеки критичного ПЗ є наявність адекватного (ефективного) нормативно-методичного забезпечення та широкомасштабне застосування інструментальних засобів підтримки процесів кваліфікаційних випробувань (експертизи), що відображають сучасний динамічний розвиток стандартизації у сфері інформаційних технологій та програмної інженерії. Основним напрямком підвищення достовірності оцінок якості ПЗ ІКС критичного застосування під час кваліфікаційних випробувань є диверсифікація технологій верифікації.

Виконання цих умов вирішальним чином визначає реальні можливості гарантування необхідного рівня безпеки та якості ІКС в цілому, в тому числі в межах risk-informed підходів до регулювання безпеки. Оцінка характеристик

якості ПЗ ІКС критичного застосування з врахуванням ризиків прихованих дефектів ПЗ є актуальною складовою реалізації risk-informed підходів до регулювання безпеки та кваліфікаційних випробувань ІКС критичного застосування в різноманітних прикладних галузях (АЕС, космос, транспорт та інше).

Верифікація програмного забезпечення на підприємстві «Радікс» проходить в декілька етапів. Це проведення статичного аналізу, огляду коду та проведення функціонального тестування з використанням ПЗ ModelSim та Quartus.

1.2.1 Проведення статичного аналізу і огляду коду

Процес проведення статичного аналізу (Static Code Analysis - SCA) і огляду коду (Code Review - CR) на мові VHDL. Даний код може описувати електронний проект (Electronic Design - ED) або його функціонально закінчені складові, призначені для конфігурації FPGA-, CPLD-, ASIC-інтегральних схем, наприклад, Functional Block Library (далі по тексту - ED). Процедура SCA і CR для коду на мові VHDL (далі по тексту - VHDL SCA / CR) використовується для перевірки відповідності аналізованого коду правилами кодування.

Процедура VHDL SCA / CR використовується з метою перевірки відсутності дефектів кодування, а також виконання при розробці електронних проектів.

Схема алгоритму проведення Static Code Analysis (SCA) і огляду коду (Code Review) представлена на Рис. 1.1.

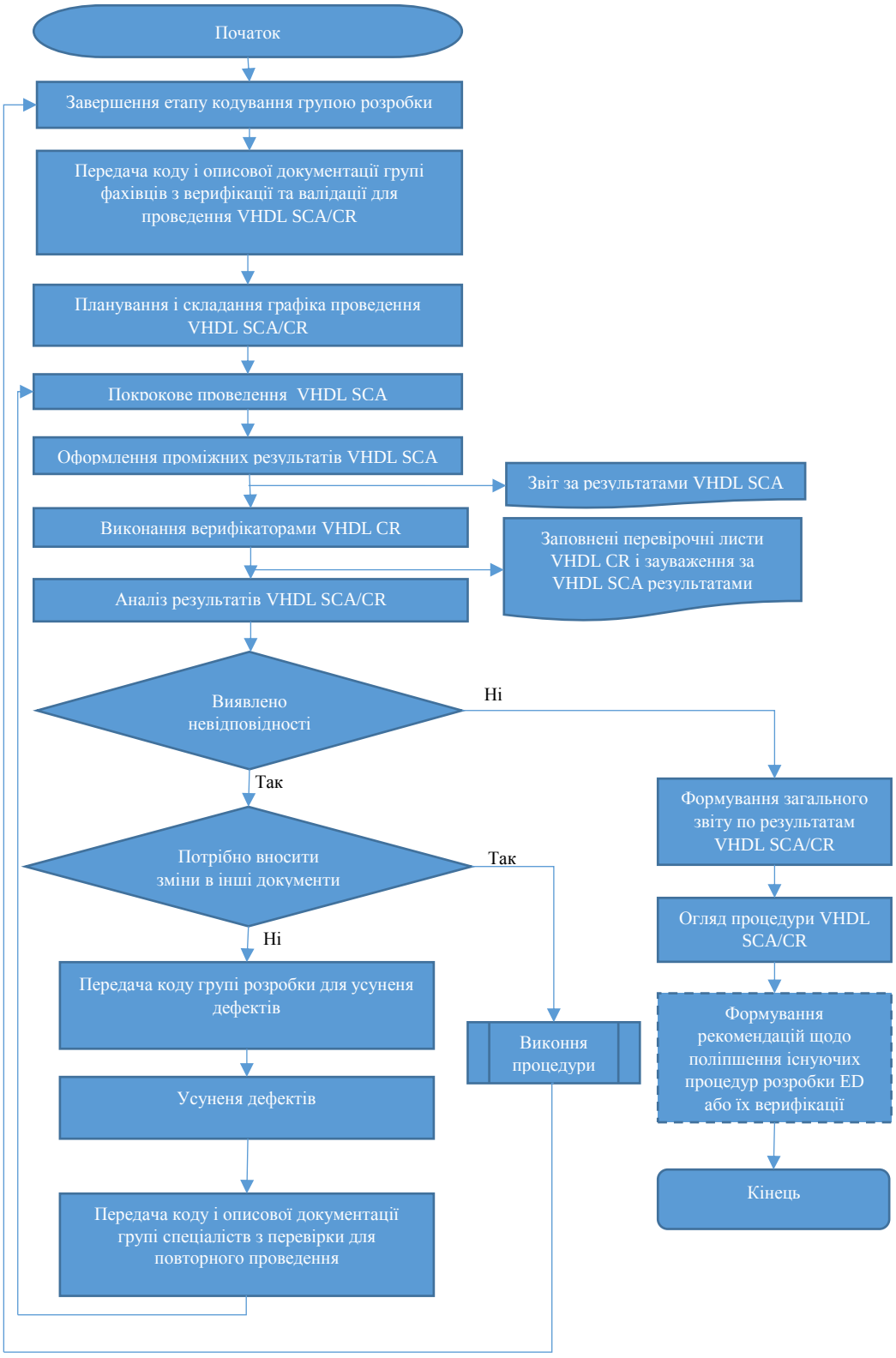


Рис. 1.1. Схема алгоритму Static Code Analysis і Code Review

Процедура VHDL SCA / CR може застосовуватися до ED, для якого завершено етап кодування.

Дана процедура може бути застосована до коду на мові VHDL, наприклад, до VHDL-коду для функціональних модулів зі складу FBL (Functional Block Library) і до коду ED мовою VHDL модулів створюваної платформи.

Склад незалежної групи фахівців з верифікації для проведення VHDL SCA / CR

Процедура VHDL SCA / CR проводиться незалежною групою фахівців з верифікації (далі - група), які не брали участь в розробці ED, володіють необхідними знаннями, навичками і досвідом в розробці (синтезі) і верифікації коду функціонально складних ED мовою VHDL.

Керівник групи верифікації несе відповідальність за виконання таких дій:

- формірованіе складу групи для проведення VHDL SCA / CR;
- планування, підготовку і проведення VHDL SCA / CR;
- гарантоване проведення даної процедури в належному порядку;
- формування звітних документів за результатами проведення VHDL SCA / CR.

Учасник групи верифікації (далі - верифікатор) відповідає за виконання VHDL SCA / CR з метою перевірки відсутності дефектів, а також виконання стандартизованих правил застосування мови VHDL при програмуванні і встановлених семантичних, пунктуаційних і синтаксичних правил кодування для підтвердження його завершеності, несуперечності або відхилення від вимог зазначених в технічному завданні та / або стандарті IEEE 1076: 2008 VHDL Language Reference Manual. Верифікатор може виконувати ролі читача (при проведенні VHDL CR) і / або реєстратора (при проведенні VHDL SCA / CR). Керівник також може виконувати роль верифікатора. На роль верифікатора не можуть призначатися співробітники, які брали участь в розробці. Перевіряється відповідність коду ED мовою VHDL. Рекомендується призначати для виконання VHDL SCA / CR не менше двох верифікаторів.

Кількість верифікаторів може бути збільшено, якщо код ED мовою VHDL відрізняється особливою складністю.

Верифікаторам повинні бути визначені обов'язки:

- читач - є головним відповідальним за читання наданих матеріалів підлягають VHDL CR; його метою є концентрація уваги на матеріалі наданого для VHDL CR і забезпечення упорядкованого потоку інформації для сприйняття іншими верифікаторами;

– реєстратор призначається для запису даних про дефекти, невідповідності і даних про проведення VHDL SCA / CR.

У проведенні VHDL SCA / CR також повинен брати участь розробник - фахівець групи розробки ED, який несе відповідальність за його створення і усунення виявлених дефектів і / або невідповідностей; під час проведення VHDL SCA / CR він відповідальний за попереднє представлення проекту ED верифікаторам.

Передача розробниками коду ED мовою VHDL групі верифікації для проведення VHDL SCA / CR.

Код ED мовою VHDL створюється в середовищі (інструментальному засобі) проектування для конкретного типу FPGA. Для такого інструментального кошти (IC) в наявності повинен бути сертифікат (ліцензії), а при необхідності і проведення апробації IC, що здійснюється в процесі вибору IC для проектування ED мовою VHDL, що не є частиною процесів VHDL SCA / CR, описаних в даній процедурі . Наприклад, для створення FPGA ED для FPGA виробництва фірми ALTERA самою фірмою-виробником було розроблено відповідне IC проектування «Quartus», яке офіційно поширюється для промислового застосування і ліцензується фірмою-виробником через веб-інтерфейс або при «коробкової» постачання, а сервісний пакет " Safety Integrity Level 3 (SIL3) Functional Safety Data Package ", який пройшов сертифікацію в незалежній органі по сертифікації - групі компаній TÜV Rheinland.

Після того, як створення (синтез, розробка), попереднє функціональне тест-моделювання та налагодження зазначеного коду ED мовою VHDL завершені, за рішенням керівника групи розробки ED керівнику групи верифікації для безпосереднього проведення VHDL SCA / CR встановленим порядком представляється (передається) наступне:

- повний комплект дійсної технічної проектної документації, використаної в якості вихідних даних при розробці коду ED мовою VHDL (в т.ч. технічне завдання на розробку продукту, концепція продукту, опис архітектури продукту);

- перелік і короткий опис функціональних бібліотек, що підключаються розробниками коду ED мовою VHDL в середу проектування ED мовою VHDL, а також опис порядку підключення (відключення) зазначених бібліотек при розробці коду, який підлягає VHDL SCA;

- затверджена керівником групи розробки ED описова документація (пояснювальна записка або опис коду ED, опис застосування коду ED, опис застосування правил кодування на мові VHDL розробниками при розробці ED, інші документи) на розроблений код ED мовою VHDL (в паперовому та / або електронному вигляді), оформлена відповідно до вимог стандарту IEEE 1016: 2009, а також діючих на підприємстві стандартів ЕСПД серії ГОСТ 19.XXX, стандартів і робочих інструкцій підприємства;

- повна точна електронна копія розробленого коду ED мовою VHDL (коду на мові VHDL універсального функціонального блоку ED) в електронному вигляді (у файлах формату * .vhd);

- повна точна електронна копія функціональних бібліотек, що підключаються розробниками коду ED мовою VHDL в середу проектування ED мовою VHDL при розробці коду, що підлягає VHDL SCA.

Попереднє подання проекту ED (наприклад, уявлення Architecture Description (AD) або Detailed Description (DD)) необов'язковий етап процедури VHDL SCA / CR, під час якого розробник дає коротку характеристику проекту ED верифікаторам до початку проведення VHDL SCA / CR. Керівник групи верифікації відповідає за визначення необхідності проведення і планування даного етапу.

Попереднє подання проекту ED має проводитися за умов великого обсягу і складності проекту ED, а також недостатнього ознайомлення з ним верифікаторів.

Попереднє подання проекту ED має проводитися завжди (в т.ч. повторно), якщо створюються нові або вносяться зміни в вихідні документи, необхідні для проведення VHDL SCA / CR.

Вхідними критеріями для попереднього подання проекту ED є:

- верифікатори повідомлені про проведення VHDL SCA / CR керівником групи верифікації;

- керівником групи верифікації визначено необхідність проведення попереднього подання проекту ED;

- група розробки ED готова надати матеріали для попереднього подання проекту ED.

Завдання, які вирішуються на стадії попереднього подання проекту ED:

1. Керівнику групи верифікації сповістити верифікаторів про майбутній нараді для проведення попереднього надання проекту ED.

2. Попереднє подання одним з фахівців групи розробки ED проекту ED.

3. Затвердити відповідність потреб стадій підготовки і проведення VHDL SCA / CR з цілями короткого представлення проекту ED (затверджується верифікаторами).

Вихідним критерієм даної стадії VHDL SCA / CR є твердження верифікаторами задовільного проведення наради за попередньою поданням проекту ED у вигляді зафіксованого керівником групи верифікації протоколу проведення даної наради.

Після отримання всієї необхідної і затвердженої відповідними посадовими особами супровідної і описової документації на розроблений код ED мовою VHDL (в паперовому та / або електронному вигляді) і повних точних електронних копій розробленого коду ED мовою VHDL, а також всіх підключається в середу проектування при розробці даного коду функціональних бібліотек (в електронному вигляді) керівник групи верифікації в суворій відповідності із загальним планом перевірки та затвердження (Overall Verification and Validation Plan) здійснює детальне пл вання і розподіляє ролі і відповідальності членів групи при безпосередньому проведенні VHDL SCA / CR.

Планування полягає в визначення об'єктів, які підлягають VHDL SCA / CR і визначення часу необхідного для проведення VHDL SCA / CR. Графік проведення

VHDL SCA / CR повинен визначати дату і час проведення VHDL SCA / CR, а також склад групи верифікації.

Вхідні критерії для планування проведення VHDL SCA / CR:

- позначені вимоги до плану проведення VHDL SCA / CR;

- за попередніми оцінками встановлений обсяг коду ED мовою VHDL;

- верифікатори пройшли професійну підготовку або визначений план їх підготовки;

- встановлено цілі і завдання для проведення VHDL SCA / CR.

Вхідні критерії для складання графіка проведення VHDL SCA / CR:

- описова документація на розроблений код ED мовою VHDL наближається до стадії завершення;

- доступні ресурси необхідні для проведення VHDL SCA / CR;

- керівник групи верифікації оповіщає верифікаторів про проведення VHDL SCA / CR.

Завдання, які вирішуються на етапі планування VHDL SCA / CR:

1. Визначити об'єкт, що підлягає VHDL SCA / CR, і правила, на правильність застосування яких буде перевірятися код ED мовою VHDL.

2. Оцінити ресурси для проведено VHDL SCA / CR і розподілити бюджет часу.

На виконання VHDL CR і проведення аналізу результатів VHDL SCA / CR рекомендується планувати час відповідно до обсягу коду - 100-150 рядків коду на годину. В процесі неодноразового виконання VHDL CR вищенаведені дані можуть бути скориговані керівником групи верифікації.

3. Встановити етапи проведення VHDL SCA / CR.

4. Визначити залежність від інших груп, що беруть участь в створенні коду ED мовою VHDL або відповідної описової документації.

Завдання, які вирішуються на етапі складання графіка проведення VHDL SCA / CR:

1. Визначити склад групи верифікації для проведення VHDL SCA / CR і розподілити обов'язки між верифікаторами.

2. Визначити дату проведення етапів VHDL SCA / CR.

3. Перевірити відповідність переданого розробниками коду ED мовою VHDL для VHDL CR вимогам, зазначеним в розділі 3 даної процедури.

4. Провести порівняльний аналіз версій компонентів ED, що підлягають процедурі VHDL SCA / CR, в разі регресійного етапу проведення SCA / CR.

5. Скласти розклад проведення VHDL SCA / CR.

Стадія планування і складання графіка буде завершена, коли виконані наступні критерії:

- складено план проведення VHDL SCA / CR;

- якщо не можна визначити точні дати деяких етапів в плані VHDL SCA / CR повинні бути вказані межі ймовірних дат;

- в плані проведення VHDL SCA / CR вказані адекватні ресурси;

- складено графік проведення VHDL CR;

- розподілені обов'язки між верифікаторами.

6. Методика статичного аналізу для коду ED мовою VHDL

Процедура VHDL SCA виконується з використанням ІС проведення VHDL SCA, яке вибирається для конкретного проекту в рамках проведення аналізу і вибору ІС, результат якого відображається у відповідному звіті. У цій процедурі описано порядок проведення VHDL SCA з використанням наступних ІС:

1) ІС HDL Designer, що дозволяє виявити невідповідності стандартизованим правилам (помилки) в ED на стадії розробки, а також в автоматичному режимі перевірити стиль кодування, що дозволяє виявити функціональні та структурні проблеми в VHDL-коді;

2) ІС Quartus II, що дозволяє провести попередній аналіз і синтез коду вбудованої утилітою "Analysis & Synthesis".

3) ІС Verification Support Tool (далі VTS), що дозволяє провести порівняльний аналіз MD5 для vhdl файлів і сформувати csv звіт за результатами порівняння. Дане ІС використовується в разі проведення регресійного SCA.

7. Методика огляду коду ED мовою VHDL

Для виконання верифікаторами VHDL CR керівник групи верифікації перевіряє наявність описової документації на розроблений код ED мовою VHDL і результатів проведення VHDL SCA, отриманих за допомогою IC HDL Designer, у верифікаторів, які на даному етапі зобов'язані підготуватися до наради з виконання аналізу результатів проведення VHDL SCA / CR.

Головна відповідальність за виконання VHDL CR покладається на верифікаторів.

Вхідними критеріями для виконання верифікаторами VHDL CR є:

- попереднє представлення проекту ED, в разі необхідності його проведення, було успішно завершено;

- описова документація на розроблений код ED мовою VHDL і безпосередньо VHDL-код передані групі верифікації;

- отримані результати VHDL SCA з використанням IC HDL Designer відповідно до методики, описаної в розділі 6;

- необхідні допоміжні матеріали доступні для виконання VHDL CR;

- кількість часу, необхідне для виконання VHDL CR було погоджено з верифікаторами.

Завдання, які вирішуються під час виконання VHDL CR:

1. Сформувати питання, відзначити невідповідності, можливі дефекти в наданому для огляду коді ED мовою VHDL для розгляду їх на нараді щодо виконання аналізу результатів проведення VHDL CR. Під час виконання даного завдання необхідно використовувати перевірки лист для VHDL CR на відповідність правилам D7.23 FSC Guideline on Design and Coding with VHDL (приклад перевірочного листа наведено в шаблоні звіту за результатами проведення VHDL SCA / CR в додатку А). В даному перевірочному аркуші необхідно поставити одну позначку в одній з колонок «Complies», «Does not comply», «N / A» (Not Applicable). У разі якщо розглянуте правило не перевіряється в файлі формату *.vhd («N / A»), необхідно дати пояснення до даного правила (чому правило не може бути перевірено).

2. Проаналізувати результати VHDL SCA шляхом огляду порушень правил виявлених IC HDL Designer.

3. Кожен верифікатор повинен зазначити кількість витраченого часу на виконання VHDL CR.

Етап виконання VHDL CR буде завершено, коли задоволені наступні критерії:

- кожен верифікатор заповнив перевірки лист для VHDL CR;

- підготовлені питання, список невідповідностей та / або дефектів виявлених в даному коді ED мовою VHDL і зауваження за результатами VHDL SCA.

Аналіз результатів VHDL SCA / CR

Протягом наради, під час якого проводиться аналіз результатів виконання VHDL SCA / CR обговорюються питання, невідповідності і / або дефекти виявлені верифікаторами, а також приймається рішення про ступінь важливості виявлених невідповідностей та / або дефектів. Під час обговорення наданих матеріалів, що підлягають VHDL SCA / CR, верифікаторами можуть бути виявлені нові невідповідності і / або дефекти, не виявлені під час проведення VHDL CR. Варіанти усунення виявлених невідповідностей та / або дефектів не повинні обговорюватися під час проведення аналізу результатів виконання VHDL SCA / CR.

Відповідальність за формування результатів виконання VHDL SCA / CR покладається на всіх верифікаторів.

Вхідними критеріями для успішного проведення наради з аналізу результатів виконання VHDL CR є:

- верифікатори, виконують VHDL CR, присутні в повному складі;

- описові документи, що стосуються розробленого коду ED мовою VHDL, були доступні для виконання VHDL CR;

- визначено мету і завдання наради з аналізу результатів виконання VHDL SCA / CR.

Завдання, які вирішуються під час проведення наради з аналізу результатів виконання Code Review:

1. Короткий вступ, здійснюваний керівником групи верифікації, під час якого він повинен:

- переконатися в тому, що всі верифікатори, виконують VHDL CR, знають свої обов'язки;

- якщо необхідно, то знову довести до відома членів групи верифікації структуру процедури VHDL SCA / CR (рис.1) і місце VHDL CR в ній.

2. Перевірка керівником групи верифікації рівня підготовленості верифікаторів, яка полягає в:

- опитуванні верифікаторів про витрачений ними часу на виконання VHDL CR, а також про кількість і основних видах виявлених при аналізі дефектів і / або невідповідностей;

- ухваленні рішення про те, що проведення наради з аналізу результатів виконання VHDL CR ґрунтується на всіх раніше підготовлених зауваженнях і охоплює всі аспекти проведеного аналізу.

3. Зачитування переліку матеріалу, наданого для VHDL CR, читачем.

4. Ідентифікація невідповідностей та / або дефектів верифікаторами:

- визначити невідповідності і / або дефекти за результатами виконання VHDL CR;

- порівняти отримані за допомогою IC HDL Designer результати VHDL SCA, і результати VHDL CR і ідентифікувати невідповідності і / або дефекти для подальшого їх усунення (визначити пріоритет (значимість) впливу на безпеку виявленої невідповідності зазначених вимог);

- оцінити ступінь впливу ідентифікованих в процесі виконання VHDL SCA / CR невідповідностей та / або дефектів на інші документи (сутності) розроблені раніше в рамках розглянутого конкретного проекту і визначити необхідність внесення змін до них;

- у разі якщо верифікаторами приймається рішення про те, що ідентифіковані в процесі виконання VHDL SCA / CR невідповідності і / або дефекти спричинять зміни в раніше створених документах (сутності) в рамках розглянутого

конкретного проекту, то керівник групи верифікації ініціює процес внесення змін відповідно до процедури описаної в МК-24;

- уникати обговорення стилю або варіантів виправлення знайдених невідповідностей та / або дефектів.

5. Фіксування всіх невідповідностей та / або дефектів реєстратором, який повинен:

- зазначити кожне знайдене невідповідність і / або дефект, вказуючи його місце в розглянутому коді ED мовою VHDL і номер порушеного правила;

- під час або після завершення наради з аналізу результатів виконання VHDL CR переглянути зафіксовані невідповідності і / або дефекти для подальшого надання їх на затвердження верифікаторам;

- зафіксувати час, витрачений на проведення наради з аналізу результатів виконання VHDL CR.

6. Визначити статус коду ED мовою VHDL, після прийняття одного з двох можливих рішень:

- якщо невідповідності і / або дефекти не виявлені, то результат VHDL CR затверджується як позитивний і керівником групи верифікації оформляється загальний для VHDL SCA / CR звіт про проведену процедуру і отриманих результатах;

- якщо невідповідності і / або дефекти виявлені, то приймається рішення про подальшу передачу коду ED мовою VHDL групі розробки ED на їх усунення, після чого організувати повторне проведення VHDL SCA / CR, а результати проміжного аналізу результатів виконання VHDL CR оформляються у вигляді проміжного звіту або чернетки підсумкового звіту.

Проведення наради з аналізу результатів виконання VHDL CR буде завершено, коли виконані наступні критерії:

- наданий код ED мовою VHDL пройшов процедуру VHDL CR в тому обсязі, в якому планувалося;

- зафіксовані невідповідності і / або дефекти з чітко вказаним їх розташування в наданому матеріалі і цей список затверджений верифікаторами;

- у разі виявлення невідповідностей та / або дефектів керівник групи верифікації передає код ED мовою VHDL і список невідповідностей та / або дефектів за результатами проведення VHDL SCA / CR групі розробки коду ED мовою VHDL для їх усунення.

Повторне VHDL SCA / CR

Вхідними критеріями для повторного проведення VHDL SCA / CR є:

- розробник завершив усунення невідповідностей та / або дефектів в коді ED мовою VHDL і передав його для повторного проведення VHDL SCA / CR;

- керівнику групи верифікації або призначеним ним верифікатором переданий код ED мовою VHDL з усуненими невідповідностями та / або дефектами.

Завдання, які вирішуються під час повторного проведення Static Code Analysis і Code Review:

1. Повторно провести VHDL SCA

2. Повторно провести VHDL CR

3. Керівнику групи верифікації завершити загальний звіт за результатами проведення VHDL SCA / CR.

Етап повторного VHDL SCA / CR завершений, коли виконані наступні критерії:

- наданий на повторне проведення VHDL SCA / CR код ED мовою VHDL пройшов цю процедуру;

- невідповідності і / або дефекти не виявлені;

- якщо були ідентифіковані нові невідповідності і / або дефекти, то їх повторно необхідно передати на усунення (з подальшим повторним проведенням VHDL SCA / CR) групі розробки коду ED мовою VHDL (в т.ч. для документального пояснення причин, за якими окремі невідповідності і / або дефекти не можуть бути усунені). Дане рішення узгоджується між керівниками групи верифікації та групи розробки коду ED мовою VHDL;

- зафіксовані результати повторного проведення VHDL SCA / CR в загальному звіті за результатами проведення VHDL SCA / CR.

Для того щоб проведення VHDL SCA / CR було завершено потрібно, щоб:

- всі ідентифіковані невідповідності і / або дефекти були усунені або задокументовані причини, за якими окремі невідповідності і / або дефекти не можуть бути усунені (рішення про прийнятність таких пояснень узгоджується між керівниками групи верифікації та групи розробки коду ED мовою VHDL і приймаються спільно);

- керівник групи верифікації завершив складання і затвердив загальний звіт за результатами проведення VHDL SCA / CR.

Огляд проведення процедури VHDL SCA / CR це стадія, під час якої класифікують причини дефектів, знайдених під час проведення VHDL SCA / CR. Ця діяльність є важливим етапом для попередження виникнення аналогічних дефектів у подальшій роботі.

Керівник групи верифікації призначає нараду з огляду результатів проведення процедури VHDL SCA / CR. Все верифікатори беруть участь в проведенні даної наради.

Завдання, які вирішуються під час наради з огляду результатів проведення процедури VHDL SCA / CR:

1. Здійснити вибірку всіх дефектів зі звіту за результатами проведення VHDL SCA / CR для обговорення.

2. Визначити потенційні причини дефектів.

3. Зафіксувати результати цієї наради. Також можуть бути сформульовані рекомендації щодо поліпшення існуючих процедур розробки коду ED мовою VHDL і / або його верифікації з подальшим їх наданням групі розробки коду ED мовою VHDL для подальшого попередження дефектів.

Порядок оформлення результатів проведення VHDL SCA / CR

Всі дії по проведенню VHDL SCA / CR повинні бути детально задокументовані і можуть бути внесені в систему стеження за вадами (bugtracking system) для обліку і моніторингу невідповідностей та / або дефектів, а також відстеження процесу усунення цих невідповідностей та / або дефектів. По завершенню проведення VHDL SCA / CR керівником групи верифікації повинен

бути оформлений документальний звіт за його результатами згідно шаблону, наведеного в додатку А (мова оформлення звіту залежить від вимог до конкретного проекту ED).

У звіті про проведення VHDL SCA / CR повинні бути вказані:

- дані про об'єкт VHDL SCA / CR;

- атрибути ІС проведення VHDL SCA;

- опис порядку проведення VHDL SCA / CR;

- порівняльний аналіз версій компонентів ED, що підлягають процедурі VHDL SCA / CR (оформляється як додаток до основного звіту. Приклад наведено в додатку Д);

- первинні і проміжні результати проведення VHDL SCA / CR (в т.ч. порушення правил, зауваження, рекомендації);

- результати оцінювання ступеня впливу ідентифікованих в процесі виконання VHDL SCA / CR невідповідностей та / або дефектів на інші документи (сутності) розроблені раніше в рамках розглянутого конкретного проекту (вказати чи потрібне внесення змін відповідно до процедури описаної в МК-24);

- результати повторного проведення VHDL SCA / CR після усунення невідповідностей (якщо такі були виявлені в процесі проведення VHDL SCA / CR);

- загальні висновки за результатами проведення VHDL SCA / CR, із зазначенням оцінки рівня відповідності аналізованого коду ED вимогам IEEE 1076: 2008 і D7.23 FSC Guideline on Design and Coding with VHDL, а також кількості виявлених, усунених і не усунених невідповідностей та / або дефектів.

Якщо виявлені невідповідності та / або дефекти не усунуто, то в звіті повинно бути зафіксовані пояснення причин залишення розробниками коду ED на зике VHDL без змін (виправлень) по кожному такому порушенню або зауваженням, завіреним підписом розробника.

Звіт повинен бути викладений у доступній формі, зрозумілою фахівцям, які не брали участі в проведенні VHDL SCA / CR.

Наступним етапом веріфікації буде проведення процесу функціонального тестування (Functional Testing - FT) коду на мові VHDL (далі по тексту - VHDL FT)

електронного проекту (Electronic Design - ED) будь-якого модуля / блоку платформи ІКС або функціонально закінченої частини для програмування FPGA -, CPLD-, ASIC-інтегральних схем - наприклад, функціонального блоку зі складу Function Block Library (FBL) - далі за текстом - ED / FBL. Процедура VHDL FT використовується для перевірки коректності реалізації всіх функціональних вимог до ED / FBL на мові VHDL, описаних в специфікаціях вимог на платформу або програмно-технічний комплекс (далі ПТК) інформаційно-керуючої системи (далі ІУС), її (його) модуль / блок , бібліотеку функціональних блоків, а також в архітектурному / детальному описі самого електронного проекту (бібліотеки функціональних блоків)[7].

Метою процедури VHDL FT є підтвердження відповідності результатів симуляції виконання реалізованого функціоналу ED / FBL очікуваних результатів, встановленим у вихідних документах, таких як план і специфікація функціонального тестування, складені на основі описових документів розробленого ED / FBL (наприклад - Architecture Description (AD) і Detailed Description (DD) для ED).

VHDL FT може застосовуватися до ED / FBL на мові VHDL, для якого завершено етап розробки програмного коду, складений план функціонального тестування (VHDL FT Test Plan) і специфікація тестів VHDL FT (VHDL FT Specification) або об'єднаний документ (ED / FBL VHDL FT Plan & Specification).

Дана процедура також може бути застосована до будь-якої частини ED мовою VHDL, наприклад, до VHDL-коду для функціональних модулів зі складу FBL.

У проведенні процедури VHDL FT бере участь незалежна група фахівців з верифікації (далі - група верифікації) на чолі з керівником групи верифікації (Verification team leader).

Обов'язки групи верифікації при проведенні процедуриVHDL FT:

- загальне планування виконання функціонального тестування коду на мові VHDL (як ED для ПЛІС модуля, так і FBL платформи або конкретного ПТК ІКС) і порівняльний аналіз змін в компонентах ТО;

- складання специфікації для проведення FT і розробка відповідного тестового (симулює) програмного забезпечення (Test Benches);

- безпосереднє проведення функціонального тестування коду ED або FBL (частини FBL) в повному обсязі;

- складання звітів (VHDL FT Report) за результатами процедури VHDL FT.

Керівник групи верифікації відповідає за:

- розробку детального плану-графіка проведення VHDL FT;

- вироблення повної специфікації функціональних тестів (переліку тестових модулів і тестових перевірок);

- розподіл завдань щодо проведення процедури VHDL FT між членами групи верифікації;

- контроль за ходом виконання процедури VHDL FT;

- оформлення результатів процедури VHDL FT, їх огляд та затвердження.

Керівник може виконувати функції верифікатори. На роль верифікатора не допускається призначати співробітників, які брали участь в розробці перевіряється коду ED / FBL на мові VHDL.

З групи верифікації керівником призначається відповідальна за складання та оформлення FT Report документа особа - реєстратор.

Схема алгоритму проведення VHDL FT представлена на Рис. 1.2 (в його блоках в дужках вказані посилання на розділи / підрозділи, де описані відповідні дії).

Під час виконання різних етапів VHDL FT відповідно до структури даної процедури (Рис. 2) за рішенням керівника групи розробки ED / FBL керівнику групи верифікації для проведення відповідних дій встановленим порядком представляється (передається) технічна документація.

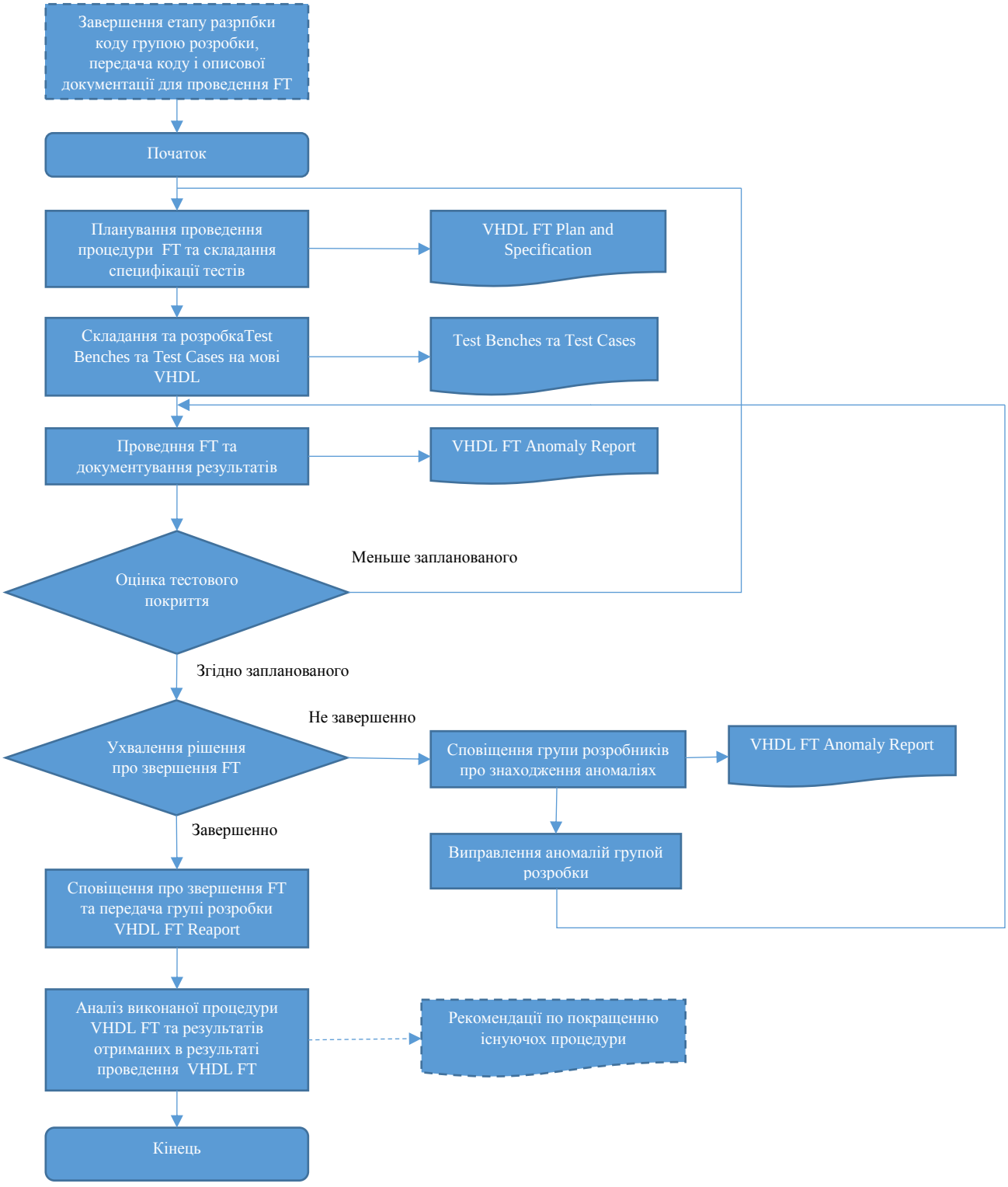


Рис. 1.2. Структура процедури Functional Testing.

Для цього необхідно передати:

- повний комплект дійсної технічної проектної документації, використаної в якості вихідних даних при розробці коду ED / FBL на мові VHDL (в т.ч. технічне завдання на розробку продукту, концепція продукту, опис архітектури продукту);

- затверджена керівником групи розробки ED / FBL описова документації (пояснювальна записка або опис коду ED / FBL, опис порядку його застосування, опис порядку застосування правил кодування на мові VHDL розробниками при розробці коду ED / FBL, інші документи) на розроблений код на мові VHDL (в паперовому та / або електронному вигляді), оформлена відповідно до вимог стандарту IEEE 1016, а також діючих на підприємстві стандартів ЕСПД і робочих інструкцій підприємства;

- повна точна електронна копія розробленого на мові VHDL коду ED / FBL.

Планування проведення VHDL FT полягає в складанні VHDL FT Test Plan & Specification. Метою створення плану проведення процедури VHDL FT є складання порядку і розкладу виконання процесу VHDL FT відповідно до вимог спільного плану перевірку, затвердження проекту (наприклад - Overall Verification and Validation Plan) і стандарту IEEE 829: 2008.

План проведення VHDL FT повинен включати в себе:

- введення, в якому формулюється мета процедури VHDL FT;

- короткий опис об'єкта, що підлягає процедурі VHDL FT;

- порівняльний аналіз версій компонентів ТО, що підлягають процедурі VHDL FT;

Примітка - при проведенні порівняння версій компонентів, що підлягають процедурі VHDL FT, повинен проводитися ретельний аналіз змін в коді ED і його складових компонент (локальних блоках і FBL) на етапі планування з метою підтвердження (непідтвердження) наявності в складі ТО компонентів, функціонал яких ідентифікований в дійсній технічної проектної документації, використаної в якості вихідних даних при розробці коду ED / FBL на мові VHDL. Для аналізу змін слід використовувати автоматизований інструмент порівняння MD5 для vhdl файлів. На етапі планування проводитися порівняння версій компонент для одного і того ж билда ТО з метою переконатися, що ТО і його компоненти не були змінені з моменту проведення процедури SCA / CR. Порівняння версій між різними тестованими білд ТО проводитися на етапі SCA / CR.

Специфікація тестів процедури VHDL FT створюється відповідно до вимог спільного плану перевірку, затвердження проекту (OVVP) і стандарту IEEE 829: 2008, повинна описувати:

- деталізацію підходу до процедури VHDL FT;

- визначення характеристик і функцій компонентів, що підлягають процедурі VHDL FT і не підлягають процедурі VHDL FT;

- визначення складу тестових наборів (Test Cases list);

- визначення критеріїв повноти тестових перевірок (Test Cases Integrity);

- визначення критеріїв виконання (невиконання) тестових наборів (Pass / Fail Test Cases);

- визначення переліку тестових симулюють (задають) модулів (Test Benches) і змінюваних в них при реалізації конкретних тестових наборів (Test Cases) параметрів (вхідних даних), які повинні використовуватися при проведенні процедури VHDL FT, із зазначенням переліків перевіряються з їх допомогою функцій і очікуваних результатів.

Тест-план і специфікація повинні відповідати таким критеріям:

- бути трассируемого до ED DD, ED AD і дизайну ED / FBL;

- бути узгодженим з ED DD, ED AD і дизайном ED / FBL;

- забезпечувати 100% покриття вимог до дизайну ED / FBL;

- бути узгодженим з інтеграційним і валідаційні тестуванням.

В рамках конкретного проекту план проведення і специфікація тестів VHDL FT можуть бути об'єднані в один документ. В такому випадку специфікація тестів VHDL FT буде містити в собі один з розділів, що охоплює планування проведення VHDL FT.

Після того, як складені план проведення і специфікація тестів VHDL FT групою верифікації проводиться огляд даних документів з метою підтвердження відповідності вимог зазначених в загальному плані перевірку, затвердження проекту (наприклад, Overall Verification and Validation Plan) і стандарті IEEE 829: 2008.

Умови необхідні для проведення огляду плану проведення та специфікації тестів VHDL FT:

- описова документація на розроблений код ED / FBL на мові VHDL, план проведення і специфікація тестів VHDL FT передані групі верифікації;

- на етапі порівняльного аналізу компонентів проекту ED не було виявлено аномалій;

- кількість часу, необхідне для виконання огляду плану проведення та специфікації тестів VHDL FT було погоджено з верифікаторами.

Для виконання верифікаторами огляду плану проведення та специфікації тестів VHDL FT керівник групи верифікації перевіряє наявність даних документів, а також наявність описової документації на розроблений код ED / FBL на мові VHDL у верифікаторів.

Виконання даного етапу проводиться шляхом заповнення верифікаторами перевірочних листів є додатками до плану проведення та специфікації тестів VHDL FT. Зазначені перевірочні листи повинні містити питання, які охоплюють всі вимоги, що висуваються до змісту і складання даних документів.

Головна відповідальність за виконання огляду плану проведення та специфікації тестів VHDL FT покладається на верифікаторів.

Test Bench - спеціальний об'єкт для проведення процедури VHDL FT, призначення якого задавати тестові впливу на тестований об'єкт і знімати отриманий результат для порівняння з очікуваним результатом. Загальна архітектура процедури VHDL FT представлена на Рис. 1.3. Test Benches повинні відповідати таким критеріям:

- містити всі необхідні входи і тестові набори (Test Cases) для впливу на тестовий об'єкт з метою виконання тестів в автоматичному режимі;

- містити всі необхідні виходи для отримання результатів роботи тестового об'єкта, і порівняння отриманих результатів з очікуваними, в результаті чого робити висновок про позитивне / негативне завершення тесту.

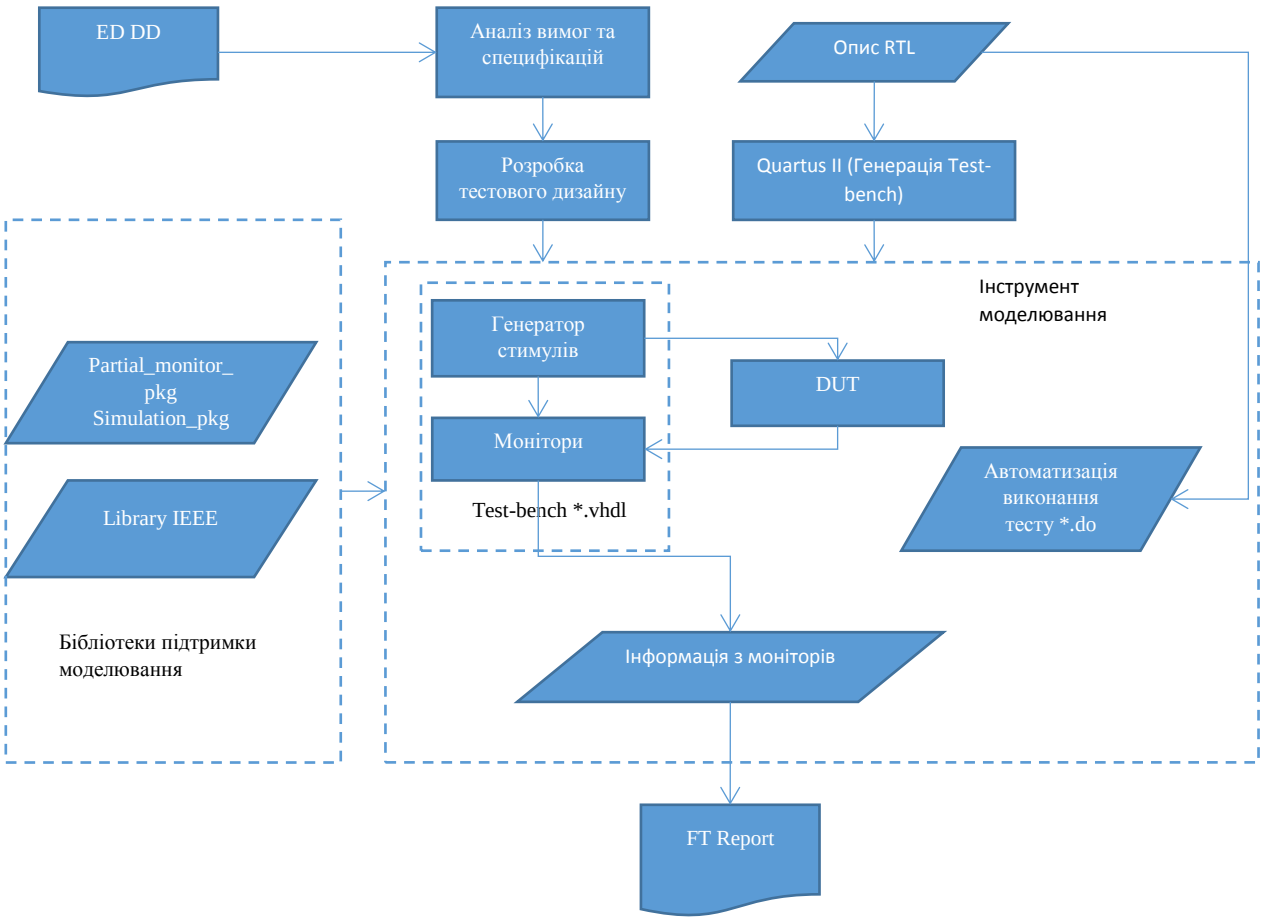


Рис. 1.3. - Загальна архітектура процедури Functional Testing

Для створення тест-Бенча на мові VHDL слід використовувати IC від фірми Altera Quartus II [6].

Безпосередньо для процедури VHDL FT крім Test Benches необхідно визначити перевірочні набори (тестові набори) - Test Cases, які будуть вхідними даними при моделюванні і тестуванні функцій ED / FBL.

Метою виконання будь-якого тестового набору є або продемонструвати наявність в коді ED / FBL дефекту, або довести його відсутність.

Основним джерелом інформації для створення Test Cases є описова документація, що стосується розробленого ED / FBL, де повинні бути визначені функціональні вимоги, які описують поведінку ТО з позицій того, що повинен виконувати об'єкт в різних ситуаціях (реакція ED / FBL на різні вхідні впливу).

Для того, щоб виявити різні дефекти у функціонуванні ED / FBL необхідно використовувати різні групи (типи) вхідних динячих в залежності від особливостей

застосування ED / FBL. Нижче наведені приклади різних груп (типів) вхідних даних.

1) Граничні значення - окремий вид допустимих значень, передача яких на вхід коду ED / FBL може виявити дефект. Зазвичай за допомогою тестування граничних умов виявляються проблеми з арифметичним порівнянням чисел або з ітераторами циклів. В даному випадку використовуються вхідні значення, що знаходяться свідомо всередині допустимого діапазону.

Один із способів перевірки стійкості функціонування ED / FBL на значеннях, близьких до граничних - створити для кожного входу як мінімум три Test Case:

- значення всередині діапазону;
- мінімальне значення;
- максимальне значення.

Для ще більшої впевненості в працездатності ED / FBL використовують п'ять Test Case:

- значення всередині діапазону;
- мінімальне значення;
- мінімальне значення + 1;
- максимальне значення;
- максимальне значення - 1.

Такий спосіб перевірки називається перевіркою на граничних значеннях. Дана перевірка дозволяє виявити проблеми, пов'язані з виходом за межі діапазону.

2) Випадкові дані. Генерація великої кількості випадкових даних в межах допустимих значень з метою виявлення випадкових заборонених комбінацій.

3) Направлене тестування. Чітко зумовлений набір даних для досягнення певного стану. Основне призначення - збільшення покриття коду.

4) Невірні дані. При перевірці поведінки ED / FBL необхідно перевіряти її поведінку під час передачі їй даних, не передбачених вимогами. Невірні дані, як і допустимі, також можна розділяти на різні класи еквівалентності. Основна мета перевірити діагностичні властивості.

5) Повторна ініціалізація системи. Механізми повторної ініціалізації ED / FBL під час його роботи також можуть містити дефекти. В першу чергу ці дефекти можуть проявлятися в тому, що не всі внутрішні дані системи, після повторної ініціалізації, прийдуть в початковий стан. В результаті може статися збій в роботі ED / FBL.

6) Класи еквівалентності. При розробці Test Cases може виникнути така ситуація, в якій різні вхідні значення призводять до одних і тих же реакцій ED / FBL. Якщо при цьому такі вхідні значення мають щось спільне, то можливе об'єднання таких значень в класи еквівалентності, тобто виконання еквівалентного розбиття множини допустимих вхідних значень. Розбиття на класи еквівалентності - в першу чергу, спосіб зменшення необхідного числа Test Cases. Розбиття на класи еквівалентності особливо корисно, коли на вхід ED / FBL може бути подано велику кількість різних значень; тестування кожного можливого значення призвело б до занадто великого обсягу тестування.

Розглянуті вище граничні умови можуть служити прикладом класів еквівалентності:

- значення з середини інтервалу;
- граничні значення;
- неприпустимі значення за межами інтервалу.

Таким чином, тестування граничних умов і робастности є окремим випадком тестування з використанням класів еквівалентності - замість того, щоб тестувати все неприпустимі значення, вибираються тільки сусідні з граничними.

При визначенні класів еквівалентності слід керуватися наступними правилами:

- завжди буде, щонайменше, два класи: коректний і некоректний;
- якщо вхідна умова визначає діапазон значень, то, як правило, буває три класи: менше ніж діапазон, всередині діапазону і більше ніж діапазон (значення на кінцях діапазону можуть трактуватися як граничні значення);
- якщо елементи діапазону обробляються по-різному, то кожному варіанту обробки будуть відповідати різні вимоги.

7) Дані для перехресного взаємодії. Застосовується в разі, коли комбінація з двох або більше сигналів (змінних, констант, подій) надають комплексний вплив на тестовий об'єкт і необхідно перевірити поведінку ТО при впливі цієї комбінації.

Розроблені Test Cases повинні забезпечувати встановлений в OVVP і плані і специфікації по проведенню процедури FT тестове покриття. У разі, якщо заплановані Test Cases не дають необхідного тестового покриття, їх кількість може бути збільшена і відповідні зміни внесені в документи планування і специфікації. Якщо заплановане тестове покриття не можна досягти технічно, повинен бути виконаний аналіз причин, що не дозволяють досягти необхідну тестове покриття. Результати аналізу повинні бути приведені (включені) в звіті.

Метрики тестового покриття:

- функціональне покриття. Тести повинні покривати всі функціональні вимоги, що пред'являються до об'єкту описової документацією. Кожному вимогу повинен відповідати як мінімум один тест;

- кодове покриття. Метрики кодового покриття:

1) покриття виразів (Statement coverage) - в результаті тестування кожен вираз (оператор) повинен виконатися хоча б 1 раз;

2) покриття шляхів (Branches coverage) - в результаті тестування повинні виконатися всі можливі сценарії (гілки), створені засобами розгалуження IF і CASE;

3) FEC Condition (Focused Expression Coverage) - дані про покриття виразів в коді враховуючи кожен незалежний вхід в вираз.

9. Огляд керівництва користувача Test Benches і верифікаційного тестування Test Benches

В рамках VHDL FT Results Review групою верифікації проводиться огляд керівництва користувача Test Benches і верифікаційного тестування Test Benches. Даний етап виконується з метою підтвердження відсутності помилок в тестових симулюють (задають) модулях, які використовувалися під час виконання VHDL FT.

Для виконання верифікаторами огляду керівництва користувача Test Benches і верифікаційного тестування Test Benches керівник групи верифікації перевіряє наявність коду Test Benches на мові VHDL (файлу в форматі * .vht), файлу в форматі * .do, а також керівництва користувача Test Benches (User Manual) .

Відповідальність за виконання огляду керівництва користувача Test Benches і верифікаційного тестування Test Benches покладається на верифікаторів.

Умови необхідні для проведення огляду керівництва користувача Test Benches і верифікаційного тестування Test Benches:

- групою верифікації успішно завершено огляд плану проведення та специфікації тестів для VHDL FT;

- групі верифікації передана дійсна описова технічна документація на розроблений код ED / FBL на мові VHDL, керівництво користувача Test Benches і VHDL-код безпосередньо Test Bench;

- завершена розробка тест-Бенч та тест-кейсів;

- завершено складання керівництва користувача Test Benches;

- визначено достатню кількість, необхідне для виконання огляду керівництва користувача і верифікаційного тестування Test Benches.

Завдання, які вирішуються під час проведення даного етапу:

1. Провести огляд керівництва користувача Test Benches шляхом заповнення верифікаторами перевірочних листів є додатками до даного документу (приклад перевірочного листа для керівництва користувача Test Benches наведено в додатку В - Приклад оформлення керівництва користувача Test Benches).

2. Виконати верифікаційного тестування Test Benches за допомогою середовища моделювання і тестування IC ModelSim і провести його ревю. Для цього начальником групи верифікації призначається відповідальна людина з групи верифікації, який не брав участі в розробці перевіряється тест-Бенча і його тест-кейсів. Після проведення перевірки верифікатором, який проводив тестування, він заповнює поле "Reviewed by:" в шапці тест-бенч файлу.

3. Отримані верифікаторами результати верифікаційного тестування Test Benches порівняти з очікуваними результатами і при наявності дефектів (відхилень) у роботі верифікованих Test Benches ідентифікувати їх.

У разі якщо при верифікаційному тестуванні Test Benches були отримані результати, що не збігаються з очікуваними, виявлені дефекти в Test Benches повинні бути усунені, після чого необхідно повторно виконати верифікаційного тестування Test Benches.

Проведення огляду керівництва користувача Test Benches і верифікаційного тестування Test Benches може вважатися завершеним, після виконання завдань 1-3 даного підрозділу.

Методика виконання VHDL FT групою верифікації.

Після завершення етапу огляду керівництва користувача Test Benches і створення Test Benches в суворій відповідності із загальним планом перевірки та затвердження (Overall Verification and Validation Plan) і складеним планом проведення і специфікацією тестів VHDL FT (Plan & Specification VHDL FT) члени групи верифікації під керівництвом керівника групи здійснюють безпосереднє проведення VHDL FT для розробленого ED ПЛІС модуля або FBL платформи (ПТК) ІКС за допомогою ІС за допомогою ІС ModelSim відповідно до наведеної нижче методикою.

Налаштування тестового середовища на основі оцінених інструментів моделювання (відповідно до керівництвом користувача інструменту моделювання). Запустити Test Benches. Записати всі результати тесту.

Послідовність дій для запуску test bench описана нижче в даній секції.

Для кожного Test Case потрібно визначити, чи пройшов тест успішно. Для кожного не успішного тесту зафіксувати результати і дані по даному Test Case.

Перевірити покриття тестового коду, отримане під час виконання тесту. Якщо покриття коду менше необхідного, виконати дії, описані в секції 8 даної процедури.

Завершення процесу FT. Рішення про завершення процесу FT визначає, чи відповідає код ED DD, ED AD і вимогам СТП 69. В рамках цього рішення повинна

бути запропонована будь-яка відповідне доопрацювання і перевірка. При прийнятті рішення про завершення процесу FT можливі наступні рішення:

а) прийняти. Код ED / FBL приймається як є і відповідає всім наступним критеріям прийому: заданий планом FT тестове покриття досягнуто; аномалії не виявлені;

б) код ED / FBL не може бути прийнятий і відправляється групі розробці на виправлення знайдених аномалій і / або помилок. Після того як аномалії були усунені, необхідно запланувати і виконати регресійне тестування.

Передача документації з описом виявлених аномалій групі розробки про проведене FT.

Огляд результатів VHDL FT

Даний етап проводиться після виконання VHDL FT незалежно від того які результати (виявлені дефекти під час виконання VHDL FT чи ні) були отримані.

Керівник групи верифікації відповідає за організацію проведення огляду результатів VHDL FT.

При виконанні огляду VHDL FT керівник групи верифікації перевіряє наявність дійсної описової документації на розроблений код ED / FBL на мові VHDL, плану проведення та специфікації тестів VHDL FT, а також звіту за результатами VHDL FT.

Аналіз повноти покриття коду ED / FBL на мові VHDL тестовими наборами (Test Cases) може виявити частину вихідного коду на мові VHDL, яка не виконувалася в ході моделювання (тестування). Ця невиконувана частина коду ED / FBL на мові VHDL може бути результатом:

- недоліків у формуванні тестових наборів (Test Cases); - в цьому випадку повинен бути розширений перелік тестових наборів (Test Cases) для забезпечення покриття пропущеної частини коду на мові VHDL при тестуванні;

- неадекватність у вимогах до реалізації ED / FBL; - в цьому випадку повинні бути модифіковані (уточнені) вимоги, розроблені і виконані додаткові тестові набори (Test Cases);

- деактивовано VHDL-коду, яка не передбачається до виконання при кожній активації; - поєднання аналізу отриманих результатів і структури тестів повинно гарантувати наявність вбудованих засобів, за допомогою яких будь-яке несанкціоноване виконання такого VHDL-коду буде надійно попереджено, ізольовано або усунуто. Для деактивовано VHDL-коду, який виконується тільки при певних конфігураціях вхідних даних (команд), повинна бути чітко визначена і встановлена за замовчуванням нормальна експлуатаційна конфігурація коду, для якої повинні бути розроблені додаткові тестові набори та тестові процедури, що задовольняють критерію повноти покриття тестами всієї структури VHDL-коду подібного ED або компонента FBL (поєднання компонентів).

– надмірності умов; - логіка роботи відповідних умовних операторів повинна бути переглянута;

- реалізації захисного VHDL-коду; - ця частина VHDL-коду використовується для запобігання виняткових ситуацій, які можуть виникнути в процесі функціонування ED / FBL.

Після виконання процедури VHDL FT Results Review начальником групи верифікації призначається проведення наради за результатами огляду VHDL FT. Протягом наради, під час якого проводиться аналіз результатів VHDL FT, обговорюються питання, невідповідності і / або дефекти виявлені верифікаторами під час проведення VHDL FT Results Review, а також приймається рішення про ступінь важливості виявлених невідповідностей та / або дефектів. Під час обговорення наданих матеріалів, що підлягають VHDL FT Results Review, верифікаторами можуть бути виявлені нові невідповідності і / або дефекти, не виявлені на попередніх етапах. Варіанти усунення виявлених невідповідностей та / або дефектів не повинні обговорюватися під час проведення аналізу результатів VHDL FT.

Відповідальність за формування результатів виконання VHDL FT Results Review покладається на всіх верифікаторів.

Умовами необхідними для успішного проведення наради з аналізу результатів VHDL FT є:

- присутність в повному складі верифікаторів, які виконували VHDL FT Results Review;

- наявність плану проведення та специфікації тестів VHDL FT, керівництва користувача відповідних Test Benches, підготовленого групою розробки проміжного звіту за результатами VHDL FT;

- чітко визначені цілі і завдання наради з аналізу результатів VHDL FT.

Завдання, які вирішуються під час проведення наради з аналізу результатів VHDL FT:

1. Визначення переліку аналізованих матеріалів, наданих для VHDL FT Results Review.

2. Оцінка невідповідностей та / або дефектів верифікаторами:

- визначити перелік аномалій і / або дефекти за результатами виконання VHDL FT Results Review (по заповненим верифікаторами перевірочним листам для VHDL FT Results Review);

- провести оцінку ступеня серйозності виявлених аномалій, в відповідності з рекомендаціями IEEE тисячі сорок чотири;

- оцінити ступінь впливу виявлених дефектів на інші документи (сутності), розроблені раніше в рамках розглянутого конкретного проекту і визначити необхідність внесення змін до них;

- у разі якщо верифікаторами приймається рішення про те, що виявлені дефекти при виконанні VHDL FT спричинять зміни в раніше створених документах (сутності) в рамках розглянутого конкретного проекту, то керівник групи верифікації може ініціювати процес внесення змін;

- уникати обговорення стилю або варіантів виправлення знайдених невідповідностей та / або дефектів.

3. Фіксація всіх невідповідностей та / або дефектів:

- кожне знайдене невідповідність і / або дефект, має бути документовано в списку аномалій звіту за результатами VHDL FT;

- під час або після завершення наради список аномалій повинен бути узгоджений з усіма верифікаторами прінімавщімі участь в процедурі VHFL FT

4. Визначити статус звіту по VHDL FT за результатами огляду його результатів, після прийняття одного з двох можливих рішень:

- якщо невідповідності і / або дефекти не виявлені, то звіт по VHDL FT затверджується як позитивний і керівником групи верифікації приймається рішення про оформлення підсумкового звіту за результатами проведення VHDL FT (версія 1.0), до складу якого буде входити додаток з перевірочним листом, що відображає результати VHDL FT Results Review;

- якщо невідповідності і / або дефекти виявлені, то приймається рішення про необхідність передачі проміжного звіту по VHDL FT групі розробки ED / FBL для усунення дефектів і / або невідповідностей, після чого необхідно організувати підготовку до повторного проведення VHDL FT.

Проведення наради з аналізу результатів VHDL FT може вважатися успішно завершеним після виконання вище вказаних завдань.

Огляд результатів проведення процедури VHDL FT це стадія, під час якої класифікують причини дефектів, знайдених під час виконання всіх етапів, визначених структурою даної процедури (рис. 1) а також определяються шляхи вдосконалення процедури VHDL FT. Ця діяльність є важливим етапом для попередження виникнення аналогічних дефектів у подальшій роботі.

Керівник групи верифікації призначає нараду з огляду результатів проведення процедури VHDL FT. Все верифікатори беруть участь в проведенні даної наради. Нарада може бути поєднане з нарадою, описаними раніше в даному розділі даної процедури.

Завдання, які вирішуються під час наради з огляду результатів проведення процедури VHDL FT:

1. Здійснити вибірку і аналіз причин різних (за типами) дефектів зі звітів за результатами проведення VHDL FT з метою обговорення найбільш часто зустрічаються з них.

2. Визначити потенційні причини виявлених типових і рідкісних дефектів.

3. Визначити можливі і \ або необхідні шляхи удосконалення процедури VHDL FT.

4. Зафіксувати документально результати цієї наради. При цьому можуть бути сформульовані рекомендації щодо поліпшення існуючих процедур розробки коду ED / FBL на мові VHDL і / або його верифікації з подальшим їх наданням групі розробки коду ED / FBL на мові VHDL, а можливо і керівництву проекту або дирекції з контролю якості та сертифікації (з метою уточнення стандартів, методик та процедур).

5. Порядок оформлення підсумкового звіту за результатами VHDL FT.

Всі дії по проведенню VHDL FT і VHDL FT Results Review повинні бути детально задокументовані і повинні бути внесені в систему стеження за вадами (bugtracking system) для обліку і моніторингу невідповідностей та / або дефектів, а також відстеження процесу їх усунення. По завершенню проведення VHDL FT керівником групи верифікації повинен бути затверджений оформлений документальний підсумковий звіт згідно шаблону, наведеного в додатку Г (мова оформлення звіту залежить від вимог керівництва конкретного проекту).

У звіті про проведення VHDL FT повинні бути вказані:

- дані про об'єкт VHDL FT;
- атрибути IC, які використовуються для проведення VHDL FT;
- опис порядку проведення VHDL FT;
- результати проведення VHDL FT (первинні, проміжні або підсумкові);
- результати повторного проведення VHDL FT після усунення невідповідностей та / або дефектів (якщо такі були виявлені в процесі проведення VHDL FT);
- результати проведення VHDL FT Results Review (включаючи історію аномалій виявлених при FT даного об'єкта);
- загальні висновки за результатами проведення VHDL FT, із зазначенням оцінки рівня відповідності результатів VHDL FT очікуваних результатів, встановленим в які планують і нормативних документах, а також кількості виявлених, усунених і не усунених невідповідностей та / або дефектів.

Всі виявлені невідповідності та / або дефекти в списку аномалій підсумкового звіту повинні супроводжується рішенням прийнятим групою розробки щодо

шляхів вирішення даної аномалії. Якщо розробниками прийнято рішення не усувати аномалію, то в звіті повинно бути зафіксовані пояснення розробників коду ED / FBL на зике VHDL причин залишення без змін (виправлень) відповідних невідповідностей (дефектів) - по кожному такому порушенню або зауваженням окремо, завірені підписом розробника, а також повинна бути вироблено і задокументовано мотивоване згоду або незгоду групи верифікації з рішенням групи розробки щодо кожної виявленої аномалії.

Звіт повинен бути викладений у доступній формі, зрозумілою фахівцям, які не брали участі в проведенні VHDL FT і VHDL FT Results Review.

Висновки :

- верифікація дозволяє своєчасно провести коригувальні та попереджувальні дії для усунення невідповідностей, що були виявленні, і відповідно уникнути або звести до мінімуму претензії зовнішніх та внутрішніх споживачів, покращити умови експлуатації та використання об’єкта верифікації;

Розділ 1.3.Оптимізація та автоматизація процесів верифікації критичних програмних систем управління економічним об’єктом

Проведемо огляд деяких популярних методів верифікації. Для вирішення завдання було обрано п’ять альтернативних методів верифікації VHDL проектів (Таблиця 1.1)

Таблиця 1.1

Альтернативні засоби верифікації VHDL проектів

№ п/п	Системи верифікації	Позначення
1	ModelSim(базовий набір бібліотек)	ModelSim
2	QuestaSim	QuestaSim
3	Quartus	Quartus
4	Формальна верифікація	Формальна верифікація
5	ModelSim + OSVVM + UVVM	ModelSim + OSVVM + UVVM

Визначимо наступні критерії для оцінки методів верифікації з урахуванням загальноприйнятої практики та стандартів:

- функціональність (functionality) - здатність вирішувати потрібний набір задач, видавати потрібні результати, здатність до взаємодії, відповідність стандартам і правилам, відповідність ПЗ наявним індустріальним стандартам, нормативним і законодавчим актам, іншим регулюючим нормам, здатність запобігати неавторизованому доступу до даних і програм;

- вартість для користувача (cost) – вартість та тип ліцензій, вартість супроводу на рік;

- зручність використання (usability) - здатність ПЗ бути зручним у навчанні та використанні, а також привабливим для користувачів;

- зручність супроводу (maintainability) - зручність проведення всіх видів діяльності, пов'язаних із супроводом програм;

- початкові інвестиції – приблизна вартість проекту на 10 робочих місць (вартість апаратного забезпечення та системного ПЗ не враховується).

Числові оцінки матриць попарних порівнянь критеріїв для засобів верифікації VHDL проектів будемо проводити базуючись на аналізі даних з сайтів виробників відновідних систем, на експертних оцінках провідних фіхівців в визначеній галузі, а також на досвіді їх впровадження в визначеній предметній області (див.Таблиця 1.2)

Таблиця 1.2

Порівняльна характеристика методів верифікації програмного забезпечення

Показник	**ModelSim**	**QuestaSim**	**Quartus**	**ModelSim + Формальна верифікація**	**ModelSim + OSVVM + UVVM**
Зручність використання	Доволі складний інтерфейс	Зручна і проста	Зручна і проста	Доволі складний інтерфейс	Доволі складний інтерфейс
Початкові інвестиції	20 000 дол	30 000 дол	45 000 дол	20 000 дол	20 000
Зручність супроводу	Досить повна підтримка	Одна з найбільш повну реалізацій	Досить повна підтримка	Підтримка лише синхрону	Одна з найбільш повну реалізацій
Функціональність	Реалізація скриптових завдань	Реалізація скриптових завдань	Реалізація скриптових завдань	Реалізація скриптових завдання	Реалізація скриптових завдань
Вартість для користувача	Платна (~2 000 дол/ліц), існує	Платна, (~3 000 дол/ліц),	Платна (~4 500 дол/ліц),, безкоштовна версія на 30 днів	Платна(2 000 дол/ліц),	Платна (2 000 дол/ліц)

Звичайно ж, методи верифікації відрізняються не тільки наведеними параметрами. Було відібрано лише ті критерії, які дійсно можуть якось вплинути на наш вибір. Додаткові критерії можуть бути й іншими. Їх кількість також може відрізнятися від обраної нами. Слід враховувати, що:

- повинна бути можливість зібрати інформацію за кожним додатковим критерієм для всіх відібраних альтернатив;

- кількість критеріїв не повинно перевищувати 4-5, щоб не збільшити трудомісткість обробки даних до нерозумних меж.

Вся підготовча робота проведена. Тепер на практиці застосуємо метод аналізу ієрархій для вибору методу верифікації. Першим кроком буде оцінка критеріїв.

Почнемо з побудови матриці попарних порівнянь для критеріїв, тобто з другого рівня ієрархії (на першому рівні наша мета - вибір методу верифікації, на третьому - альтернативи). Для цього будуємо матрицю розмірністю 5х5 (по числу критеріїв) і підпишемо рядки і стовпці найменуваннями порівнюваних критеріїв.

Заповнюємо таблицю. Для цього попарно порівнюємо критерій із рядка з критерієм стовпця по відношенню до мети - вибору методу верифікації. Значення шкали відносної важливості вписуємо в клітинці, утвореній перетинанням відповідного рядка і стовпця.

Наприклад: ми вважаємо, що при виборі методу верифікації вартість має істотну перевагу перед зручність супроводу. В таблиці 1.3. з цією оцінкою відповідає значення "5". Тому у клітинці на перетині рядка " Вартість для користувача " та стовпця " Зручність використання " я записую значення "5".

Очевидно, що діагональ цієї матриці буде заповнена значенням "1", а клітинки, що лежать нижче діагоналі будуть заповнені зворотними значеннями. Отже, у комірці на перетині рядка " Зручність використання " та стовпця "Вартість для користувача " я записую значення "1/5". І так далі для кожної пари критеріїв.

Таблиця 1.3

Критерії оцінювання засобів верифікації VHDL проектів

Критерії	Початкові інвестиції	Зручність супроводу	Зручність використання	Функціональність	Вартість для користувача	Оцінки компонент власного вектора	Нормалізовані оцінки вектора пріоритету
Початкові інвестиції	1	2	1/2	1/5	1/8	0,47818	0,06179
Зручність супроводу	1/2	1	1/6	1/8	1/8	0,26481	0,03422
Зручність використання	2	6	1	1/3	1/5	0,95635	0,12357
Функціональність	5	8	3	1	1/3	2,09128	0,27022
Вартість для користувача	8	8	5	3	1	3,94870	0,51021

Спочатку визначаємо оцінки компонент власного вектора. Так для критерію "Вартість для користувача" це буде:

(8 x 8 x 5 x 3 x 1)1/5 = 3,94870

Отримавши суму оцінок власних векторів (=7,73931), обчислюємо нормалізовані оцінки вектора пріоритету для кожного критерію, розділивши значення оцінки власного вектора на цю суму. Для того ж критерію "Вартість для користувача" маємо:

3,94870/ 7, 73931= 0,51.

Порівнюючи нормалізовані оцінки вектора пріоритету (Таблиця 1.4) можна зробити висновок, що найбільше значення при виборі методу верифікації ми надаємо критерію "Вартість для користувача".

Необхідно перевірити, наскільки мої думки були несуперечливими при складанні матриці попарних порівнянь критеріїв. Для цього необхідно розрахувати індекс узгодженості для цієї матриці. Розділивши його на число, відповідне випадковій узгодженості матриці п'ятого порядку, що дорівнює 1,12 отримаємо відношення узгодженості (ВУ).

Таблиця 1.4

Нормалізовані оцінки вектора пріоритету критеріїв

Початкові інвестиції	0,05665
Зручність супроводу	0,03402
Зручність використання	0,13326
Функціональність	0,26870
Вартість для користувача	0,50736

У даному випадку:

ВУ = 5,90% < 10%, тобто переглядати свої судження немає потреби.

Проведемо числові оцінки матриці попарних порівнянь для критерію «початкові інвестиції».

Для оцінки розміру початкових інвестицій скористаємося даними з сайтів виробників відновідних систем, на експертних оцінках провідних фіхівців в визначеній галузі, а також на досвіді їх впровадження в визначеній предметній області

Таблиця 1.5

Початкові інвестиції в методах верифікації

Метод верифікації	Початкові інвестиції (грн.)
ModelSim	50000
QuestaSim	75000
Quartus	112000
Формальна верифікація	50000
ModelSim + OSVVM + UVVM	50000

Будуємо матрицю порівнянь (Таблиця 1.6), для чого попарно порівнюємо альтернативу з рядка з альтернативою стовпця. Ніякі інші критерії при цьому не враховуємо. Значення шкали відносної важливості вписуємо в клітинки, утворені перетинанням відповідного рядка і стовпця.

Діагональ цієї матриці заповнюємо значенням "1", а клітинки, що лежать нижче діагоналі - зворотними значеннями.

Відносна узгодженість матриці – 8,82%, тобто <10%.

Аналогічним чином будуємо матриці порівнянь для інших критеріїв та обчислюємо значення узгодженість матриці (див.додаток А).

Результати заносимо у таблицю 1.7.

У верхній рядок переносимо з таблиці 1.3 значення вектора пріоритету для кожного критерію.

Для кожної з альтернатив заповнюємо стовпці критеріїв значеннями локальних векторів пріоритету, отриманих відповідно в таблицях 3.1.3, А.1-А.4.

Підраховуємо значення глобального пріоритету для кожної з альтернатив як суму добутків значення вектора пріоритету для критерію і значення вектора локального пріоритету цієї альтернативи щодо даного критерію.

Таблиця 1.6

Матриця попарних порівнянь за критерієм «Початкові інвестиції»

Альтернативи	ModelSim	QuestaSim	Quartus	Формальна верифікація	ModelSim + OSVVM + UVVM	Оцінки компонент власного вектора	Нормалізовані оцінки вектора пріоритету
ModelSim	1	2	7	9	2	3,021900	0,413406
QuestaSim	1/2	1	6	8	1	1,888175	0,258308
Quartus	1/7	1/6	1	1/2	1/7	0,279334	0,038214
Формальна верифікація	1/9	1/8	2	1	1/2	0,425142	0,058161
ModelSim + OSVVM + UVVM	1	1	7	2	1	1,695218	0,231911
Сума	2,7540	4,2917	23,0000	20,5000	4,6429	7,309769	

Таблиця 1.7

Результати вибору альтернатив для засобів верифікації VHDL проектів

Альтернативи	Критерії					Глобальні пріоритети
	Початкові інвестиції	Можливість модернізації	Доступність	Функціональність	Вартість для користувача	
	Чисельне значення вектора пріоритету					
	0,061785	0,034216	0,123571	0,270215	0,510213	
ModelSim	0,413406	0,283109	0,145561	0,053632	0,455031	0,299871
QuestaSim	0,258308	0,211805	0,228255	0,034591	0,133456	0,128850
Quartus	0,038214	0,036197	0,058608	0,295324	0,035053	0,108527
Формальна верифікація	0,058161	0,054112	0,058608	0,245873	0,048516	0,103879
ModelSim + OSVVM + UVVM	0,231911	0,414777	0,508967	0,370580	0,327944	0,358872

Обраною альтернативою вважається альтернатива з максимальним значенням глобального пріоритету. В даному випадку це метод верифікації ModelSim + OSVVM + UVVM, на якій слід зупинити свій вибір.

Висновки

Незалежна верифікація є ключовою методикою кваліфікаційних випробувань критичного ПЗ. Її проведення є обов’язковою нормативною вимогою в сферах критичної діяльності, таких як атомна енергетика.

Метод верифікації ModelSim + OSVVM + UVVM є найкращою альтернативою при перевірці критичного програмного забезпечення для FPGA.

Розділ 2. ДОСЛІДЖЕННЯ ДІЯЛЬНОСТІ ТА СТАНУ БІЗНЕС КРИТИЧНИХ ПРОГРАМНИХ СИСТЕМ УПРАВЛІННЯ ЕКОНОМІЧНИХ ОБ’ЄКТІВ

2.1. Діагностика стану діяльності бізнес критичних програмних систем управління економічним об’єктом

Для того, щоб розібратися з діагностикою стану діяльності та актуалізації бізнес критичних програмних систем управління економічним об’єктом, потрібно розібратися з такими поняттями як інформаційні та керуючі системи, види систем, програмно-технічні комплекси та іншими.

Інформаційні та керуючі системи (ІКС) - збірне поняття, що об'єднує:

інформаційні системи, призначені для отримання, обробки, зберігання, передачі, відображення, реєстрації даних про стан і (або) функціонування систем, елементів конструкцій контрольованого об'єкта;

керуючі системи, які ініціюють спрацьовування технологічних систем при порушеннях заданих проектних меж або умов експлуатації та (або) безпосередньо впливають на технологічне обладнання з метою зміни стану або функціонування керованого об'єкта, в тому числі спрямованого на усунення порушень.

Термін «інформаційні та керуючі системи», вперше запропонований в нормах і правилах, з тих пір широко використовується в Україні - в нормативних документах, технічній документації, а також в практиці спілкування фахівців і науково-технічних публікаціях. Він відповідає загальноприйнятому в зарубіжній практиці терміну «Instrumentation and Control Systems» (I & C), який застосовується в міжнародних стандартах і літературі для систем, заснованих на електричної і (або) електронній та (або) програмованої електронної технології, які виконують функції управління, контролю і (або) спостереження за певною частиною технологічного процесу, а також функції обслуговування та спостереження, пов'язані безпосередньо з роботою самої системи .

Основні (інформаційні та керуючі) функції ІКС визначаються її призначенням. Додаткові функції ІКС повинні сприяти досягненню необхідної якості, надійності, стійкості і (або) незалежності виконання основних функцій.

До основних інформаційних функцій ІКС відносяться функції моніторингу, архівування, відображення, сигналізації і реєстрації.

Функції моніторингу забезпечують отримання даних для керуючих систем і оперативного персоналу у всіх робочих режимах енергоблоку, а також інформації, необхідної для управління аваріями і ліквідації їх наслідків (післяаварійний моніторинг).

функції архівування результатів моніторингу передбачають запам'ятовування (в хронологічній послідовності) і зберігання даних про контрольовані параметри, стан технологічних систем і устаткування, порушеннях проектних меж і умов нормальної і безпечної експлуатації, інші виникаючі події, а також дії керуючих систем і оперативного персоналу. Дані з архіву використовуються для оцінки стану енергоблоку, виявлення короткострокових і довгострокових змін (трендів) контрольованих параметрів, складання звітів, а також для подальшого аналізу виникнення, розвитку та усунення порушень і аварій.

Функції відображення і сигналізації дозволяють оперативному персоналу своєчасно виявляти порушення нормальної або безпечної експлуатації і спостерігати за результатами роботи управляючих систем та (або) власних дій, спрямованих на їх усунення. Відображення даних післяаварійного моніторингу забезпечує персонал АЕС і сторонніх експертів з безпеки, які здійснюють управління аварією і ліквідацією її наслідків, необхідними відомостями про виникнення аварійної ситуації, дії керуючих систем і операторів, розвитку аварії, стан конструкцій, систем і елементів енергоблоку під час і після проектної та запроектної аварії.

Функції реєстрації в даний час використовуються головним чином для автоматичного складання звітів передбаченої форми на паперових носіях.

До *основних керуючих функцій ІКС відносяться функції регулювання, дискретного управління, обмеження, захисту, блокування.*

Функції регулювання в експлуатаційних режимах енергоблоку забезпечують автоматичну підтримку значень технологічних та інших параметрів на рівні встановлених для них значень і (або) зменшують величину і швидкість можливих

відхилень параметрів, викликаних дією вихідних подій та перехідних процесів. У передаварійних ситуаціях, під час проектних аварій і в післяаварійних режимах функції регулювання можуть бути затребувані для стабілізації окремих параметрів енергоблоку, технологічних систем і устаткування.

Функції дискретного управління забезпечують переклад технологічного обладнання з одного дискретного стану в інший (включення, виключення, запуск, зупинка і т. п.), здійснюваний за жорстким часовим графіком (тимчасове управління) або в залежності від зовнішніх подій, стану іншого обладнання та результатів виконання попередніх дій (логічне управління).

Функції обмеження реалізуються при певних порушеннях умов нормальної експлуатації і передбачають заборону збільшення потужності реактора, обмеження або примусове зменшення потужності та інші попереджувальні дії, що дозволяють знизити частоту спрацьовування функцій захисту.

Функції захисту забезпечують своєчасне виявлення порушень експлутаційних меж і умов нормальної експлуатації, автоматичне включення відповідних виконавчих систем, викликають зупинку реактора, аварійне охолодження активної зони і відведення залишкового тепла, локалізацію радіоактивних матеріалів і обмеження аварійних викидів. Мета функцій захисту - запобігати порушенням меж безпечної експлуатації, а в разі виникнення аварій - зменшувати тяжкість їх наслідків до рівня, передбаченого проектом.

Функції блокування забезпечують запобігання важких наслідків, які можуть бути викликані незадовільним станом технологічного обладнання, порушеннями робочого режиму, виникненням інших умов, небезпечних для роботи обладнання, помилковими діями персоналу енергоблоку та ін.

До додаткових відносяться допоміжні та сервісні функції ІКС.

Допоміжні функції ІКС підтримують дії оперативного персоналу, пов'язані з дистанційним управлінням виконавчими елементами технологічного обладнання; здійснюють безперервний автоматичний контроль технічного стану самої системи, сполучених пристроїв і ліній зв'язку, включаючи виявлення та сигналізацію несправностей; забезпечують керування доступом до ресурсів системи,

реконфігурацію та відновлення функціонування при відмовах і т. п.

Сервісні функції ІКС використовуються в тих випадках, коли безпосередньо під час роботи системи необхідно змінити задані значення (уставки регулювання, сигналізації, обмежень, захистів, блокувань і т. П.), а також для автоматизації періодичних перевірок, управління конфігурацією системи і ін.

Діючі в Україні нормативні документи поділяють всі системи АЕС на системи нормальної експлуатації та системи безпеки.

ІКС нормальної експлуатації виконують зазначені вище інформаційні функції моніторингу, архівування, відображення, сигналізації і реєстрації та (або) керуючі функції регулювання, дискретного управління, обмеження, блокування, необхідні для того, щоб утримувати технологічні параметри в заданих робочих межах, змінювати режими роботи енергоблоку, запобігати порушенням експлуатаційних меж або усувати викликані ними наслідки. Під час розвитку аварії і в післяаварійному режимі системи нормальної експлуатації можуть використовуватися для отримання інформації, що дозволяє персоналу енергоблоку оцінювати стан технологічних систем і устаткування, контролювати перебіг аварії, приймати рішення по управлінню аварією та усуненню її наслідків.

ІКС безпеки виконують функції, необхідні для того, щоб запобігти переростанню аварійної ситуації в аварію, ліквідувати аварію і повернути реакторну установку в контрольований стан або обмежити наслідки аварії. До таких функцій відносяться:

інформаційні функції моніторингу, архівування, сигналізації і відображення значень контрольованих параметрів, вихідних подій, команд управління і дій персоналу;

керуючі функції захисту, автоматичного (аварійного) регулювання і дискретного управління.

Робота систем безпеки затребувана в тих випадках, коли системи нормальної експлуатації не в змозі утримувати контрольовані параметри в встановлених проектом експлуатаційних межах, наприклад внаслідок відмови, або якщо потрібна швидка і надійна реакція на перевищення експлуатаційних меж або умов безпечної

експлуатації. Виявивши таку ситуацію («умова спрацьовування»), ІКС безпеки:

видає команди управління, які ініціюють автоматичне виконання іншими інформаційними і керуючими системами, технологічними системами або обладнанням тих захисних дій, які передбачені для цього умови спрацьовування;

забороняє або відключає будь-які дії систем нормальної експлуатації і оперативного персоналу, які не збігаються з виконуваними захисними діями;

забезпечує оперативний персонал можливість спостерігати за роботою систем безпеки і, в разі необхідності, вручну виконувати дозволені дії щодо забезпечення безпеки (дублювати, ініціювати або блокувати автоматично видаються команди, запускати нові функції безпеки іт.п.).

При проектуванні ІКС визначаються функції, які вона повинна виконувати, а також обумовлені ними кордону системи, її зовнішні зв'язки та взаємодія з технологічними системами і обладнанням енергоблоку і з іншими ІКС. Істотні відмінності складових виконуваних функцій призводять до того, що складність ІКС варіюється в широких межах - від одноконтурних систем автоматичного регулювання до просторово розпорошених багатопроцесорних обчислювальних систем. При цьому можливості сучасних інформаційних технологій і доступна для застосування елементна база дозволяють поєднувати в одній системі різноманітні інформаційні та керуючі функції, характерні як для систем нормальної експлуатації, так і для систем безпеки.

До складу ІКС (Рис. 2.1) входять всі компоненти, необхідні для реалізації запропонованих їй основних і додаткових функцій: програмно-технічний комплекс, який утворює нейтральну частину системи, датчики та пертворювачі, засоби інтерфейсу «людина-машина» (ІЛМ), а також з’єднувальні кабелі, допоміжні вироби (джерела вторинного електроживлення і т. п.), сервісне обладнання і програмне забезпечення.

У керуючих системах крім зазначених компонентів передбачені також виконавчі пристрої, які безпосередньо впливають на пускові або регулюючі елементи технологічних систем і устаткування. Окремі компоненти ІКС - нормують перетворювачі, засоби ІЛМ, сервісне обладнання, програмне

забезпечення - можуть входити безпосередньо в склад системи або відноситися до її центральної частини.

Кожна ІКС призначена для конкретного технологічного об'єкта і, як правило, не може бути повторена без будь-яких змін для інших об'єктів. Це обумовлено:

відмінностями в технології і обладнанні об'єктів контролю і управління, наприклад для енергоблоків українських АЕС - типом ядерного реактора (ВВЕР-440, ВВЕР-1000) і проектом ядерної установки (В-213, В-302, В-338, В-320);

відмінностями, що склалися в результаті попередніх модернізацій суміжних інформаційних і керуючих систем, з якими повинна взаємодіяти ІКС;

необхідністю заміни деякої частини наявних периферійних пристроїв (датчиків, виконавчих механізмів, рідше - сполучних кабелів), які фізично і морально застаріли;

постановкою нових завдань, установкою більш високих вимог до властивостей системи, що випливають із загальної тенденції розвитку інформаційних технологій, результатів науково-технічних досліджень і розробок, досвід експлуатації аналогів, нових нормативно-правових актів, міжнародних стандартів і т. п.

В результаті кожна нова система, як правило, відрізняється від попередніх аналогів складом апаратних засобів і програмного забезпечення, кількістю і типами вхідних / вихідних сигналів і інтерфейсів, елементної і конструктивної базою і т. п.

Вихідні дані і вимоги до системи задає замовник (експлуатуюча організація) з урахуванням призначення ІКС, її зв'язків з іншими системами, особливостей роботи оперативного персоналу, результатів аналізу безпеки енергоблоку. Вони не повинні залежати від можливих способів реалізації системи.

Як мінімум, вихідні дані і вимоги визначають для нової або модернізованої системи:

призначення, основні функції та їх роль в забезпеченні безпеки;

переліки контрольованих параметрів, подій і станів;

діапазони зміни і межі можливих значень контрольованих параметрів;

залежності між входами і виходами кожної функції, задані у вигляді

словесних описів, формул, таблиць або алгоритмів;

вимоги до точності, тимчасовим характеристикам, надійності (безвідмовності і ремонтопридатності) кожної з основних функцій;

сигнали, інтерфейси і протоколи обміну даними з іншими інформаційними і керуючими системами.

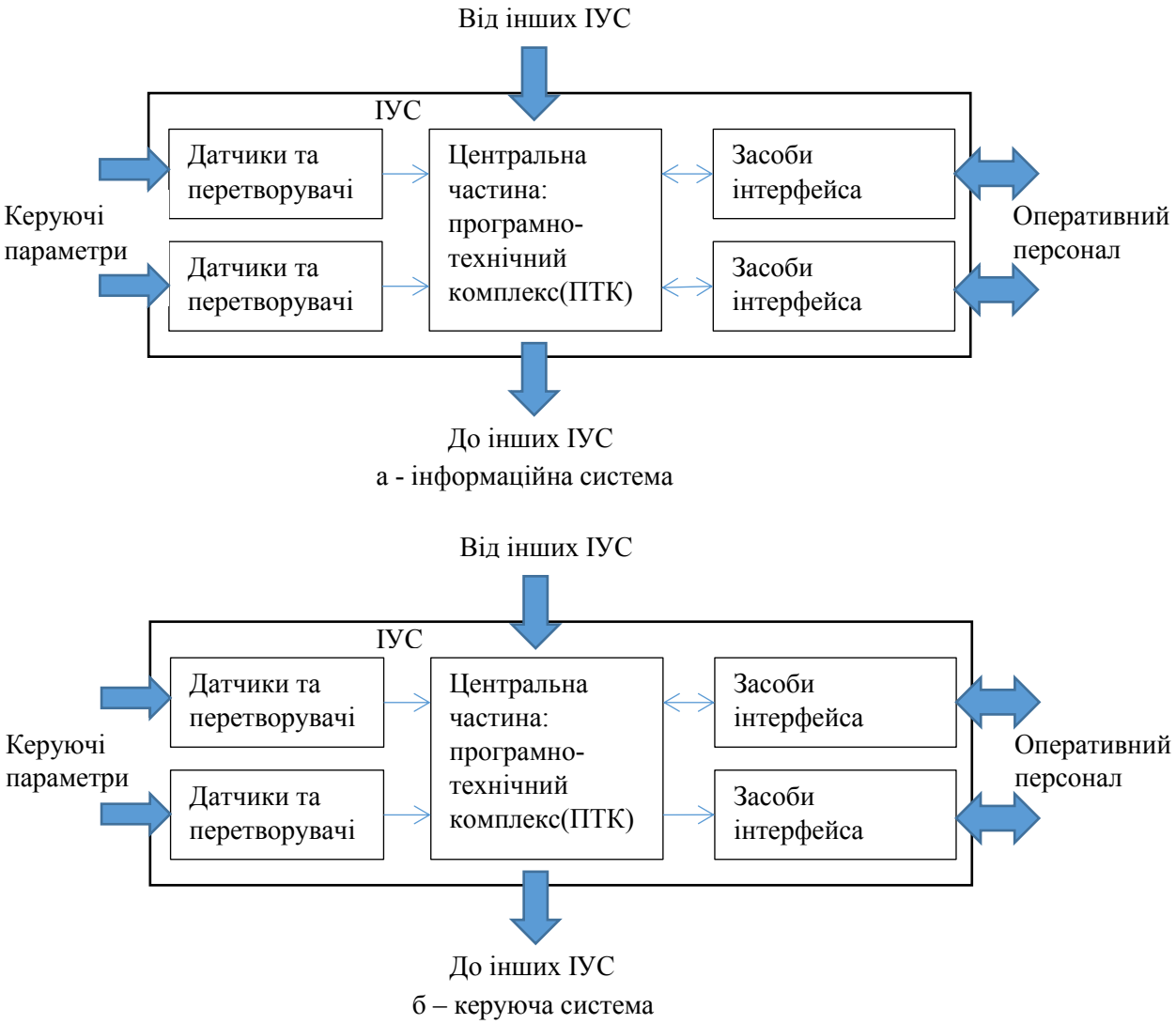


Рис. 2.1. Структурні схеми інформаційно керуючої системи

Вказуються також очікувані умови в місцях передбачуваного розміщення обладнання: робочі і граничні значення факторів, що впливають на довкілля і спеціальні середовища (при наявності), механічних та електричних впливів, якісні ознаки електромагнітної обстановки, можливі довготривалі та короткочасні відхилення параметрів електроживлення, в межах яких обладнання має виконувати

свої функції з необхідною якістю і надійністю.

Задані замовником вихідні дані і вимоги до системи конкретизують і доповнюють в технічному завданні (ТЗ) на створення або модернізацію ІКС, розробка якого завершує перший етап проектування системи.

Технічне завдання (Terms of Reference або Requirement Specification): визначає структуру системи і розподіл функцій між її складовими частинами (підсистемами);

регламентує вихідні дані і вимоги до компонентів (програмно-технічних комплексів, технічних засобів автоматизації та програмного забезпечення), які повинні бути розроблені для системи;

визначає перелік раніше розроблених («готових») компонентів, які передбачається застосувати в системі, встановлює вимоги до таких компонентів, а також необхідність, обсяг і способи перевірки відповідності цим вимогам (кваліфікації).

Кваліфікація повинна показати, що виріб, передбачуваний для застосування, здатний протягом заданого часу і в будь-яких передбачених умовах експлуатації відповідати всім вимогам, встановленим при проектуванні системи. Процедура кваліфікації базується на міжнародному стандарті, особливості кваліфікації компонентів інформаційних і керуючих систем, важливих для безпеки АЕС.

На наступних етапах проектування, у міру підвищення рівня деталізації, проводиться оцінка отриманих результатів; при цьому можуть вноситися необхідні зміни в раніше прийняті технічні рішення до тих пір, поки шляхом послідовних наближень що не отримана конфігурація системи, відповідними нормами і правилами ядерної і радіаційної безпеки, вихідними даними і вимогами замовника (експлуатуючої організації) і вимогам ТЗ на створення (модернізацію) системи. Після того, як процес проектування досяг такого рівня деталізації, коли відомо, яким чином будуть виконуватися задані функції, які нові розроблені спеціально для проектуємої системи), тиражовані і загальнопромислові компоненти будуть застосовані і як повинні бути налаштовані ці компоненти, випускається проектна документація, включаючи замовлення специфікації на поставку обладнання. На

цьому етапі передбачається інформування органу державного регулювання ядерної безпеки: результати кваліфікації, що підтверджують безпеку готових апаратних і програмних засобів, які застосовані в проектованій системі.

Після завершення проектування проводиться комплектація системи, що передбачає розробку, виготовлення і поставку нових компонентів, виготовлення і поставку тиражованих і придбання загальнопромислових компонентів. Дана стадія закінчується інтеграцією компонентів з'єднанням компонентів відповідно до проектної документації і перевіркою сумісності та інсталяцією (приведенням у дію) всієї системи. На наступних етапах проводять попередні випробування системи, які повинні підтвердити, що кожна з функцій, заданих в ТЗ, виконується з необхідною якістю і надійністю. Остаточно відповідність ІКС нормам і правилам ядерної та радіаційної безпеки і вимогам ТЗ підтверджують в процесі дослідно-промислової експлуатації та приймальних випробувань системи.

Програмно-технічний комплекс (ПТК) називають сукупність технічних засобів, що поставляються комплектно з програмним забезпеченням, необхідним сервісним обладнанням та експлуатаційноїю документацією, яка утворює центральну частину ІКС. Згадки про програмно-технічні комплекси зустрічалися в літературі і нормативній документації ще на початку 1980-х років, але лише після розробки та впровадження норм і правил це поняття стало широко застосовуватися у вітчизняній практиці, особливо при створенні і модернізації інформаційних і управляючих систем в атомній енергетиці.

Зазвичай до складу кожної ІКС входить один ПТК, який є головним компонентом системи, що беруть участь в реалізації всіх її основних і додаткових функцій. Рідше ці функції розподіляються між декількома ПТК, що входять до складу однієї системи; наприклад, деякі додаткові функції, такі як технічне діагностування, можуть виконуватися спеціально призначеними для цього програмно-технічними комплексами. Для підвищення надійності передбачаються кілька каналів в одному ПТК і (або) декілька ПТК в одній системі, які одночасно і незалежно беруть участь в реалізації основних функцій, взаємно резервуючи один одного.

Кожен ПТК виконаний у вигляді сукупності експлуатаційно-автономних складових частин, що з'єднуються на місці експлуатації електричними або іншими (наприклад, волоконно-оптичними) сполучними лініями, і під керівництвом свого («вбудованого») програмного забезпечення в загальному випадку здійснює:

прийом і перетворення в цифрову форму сигналів від периферійних пристроїв;

прийом повідомлень від інших ПТК по каналах передачі даних;

обробку прийнятої інформації і вироблення команд управління;

формування і видачу керуючих сигналів на виконавчі пристрої;

діагностування стану власного і сполученого устаткування;

архівування, відображення, реєстрацію поточної і ретроспективної інформації;

підготовку та видачу повідомлень в канали передачі даних для інших ПТК.

Звернемо увагу на наступні основні відмінності між ІКС та ПТК.

ІКС збирається безпосередньо на місці експлуатації з окремих компонентів. Замовлення компонентів, їх розміщення, з'єднання і підключення зовнішніх ланцюгів виробляються на підставі проектної документації. Функціонування ІКС може бути перевірено тільки на місці експлуатації після інтеграції всіх компонентів. Гарантії постачальників поширюються на кожен з компонентів, але не на всю систему. При модернізації зазвичай замінюється лише частина обладнання ІКС (зазвичай центральна).

На відміну від ІКС, кожен ПТК є виробом, тобто одиницею продукції виробничо-технічного призначення, налагодженої, перевіреної в заводських умовах і поставляється споживачеві комплектно з програмним забезпеченням, необхідним сервісним обладнанням та експлуатаційної документації.

Високий ступінь заводської готовності істотно спрощує і прискорює монтажно-налагоджувальні операції перед пуском енергоблоку і гарантує відповідність ПТК всім вимогам, що пред'являються. ПТК компонується на підприємстві-виробнику з покупних електротехнічних виробів та електронних елементів загального призначення, а також вбудованих складальних одиниць, не

призначених для експлуатаційно-автономного застосування, які встановлюються в несучі конструкції (оболонки). Комплектація, збірка і монтаж ПТК здійснюються на підставі конструкторської документації. Повна перевірка функціонування ПТК імітаторами сполученого обладнання здійснюється на підприємстві виробника до поставки замовнику; передбачається додаткова перевірка (передпускові випробування) на місці експлтуатації, перед інтеграцією ПТК з іншими частинами ІУС. Гарантії постачальника поширюються на ПТК в цілому, включаючи застосовані в ньому покупні комплектуючі вироби. При модернізації ІКС зазвичай замінюється весь ПТК.

Є відмінності і в програмному забезпеченні. Програмне забезпечення ІКС інтерпретується як сукупність програм, які знаходяться в постійній пам'яті ПТК і всіх «інтелектуальних» периферійних виробів (при наявності), а також сервісних програм на машинних носіях, призначених для налагодження і перевірки ІКС та її компонентів, і відповідної програмної документації.

Таким чином компоненти програмного забезпечення ІКС розробляють незалежно один від одного різні організації і в різний час. Інтеграція компонентів (вбудованих в відповідні апаратні засоби і програм на машинних носіях) і подальша перевірка програмного забезпечення може здійснюватися тільки на місці експлуатації. програмне забезпепечення ПТК є його невід'ємною частиною; найчастіше воно розробляється тією самою організацією, що і ПТК.

В процесі розробки передбачається верифікація програмного забезпечення, включаючи перевірку після його інтеграції з апаратними засобами. Запис програм в постійну пам'ять складових частин ПТК і подальша перевірка функціонування здійснюються на підприємстві-виробнику до поставки замовнику. Завантаження або модифікація програм ПТК в процесі інтеграції і при інсталяції системи зазвичай не потрібно.

Зауважимо, що за кордоном поняття «програмно-технічний комплекс» взагалі не використовують: вироби, які, за всіма ознаками, повинні розглядатися як ПТК, носять ту саму назву (Instrumentation and Control Systems), що і системи, компонентами яких вони є. Це призводить до певних проблем при встановленні

вимог та оцінки безпеки таких виробів, що імпортуються в Україну. Крім того, вказана методична невідповідність вимагає адаптації міжнародних стандартів до більш коректної системі понять, що склалася у вітчизняній практиці.

Як правило, ПТК є поодинокими виробами: кожен программнотехніческій комплекс розробляється, виготовляється і поставляється в якості компонента конкретної ІКС за разовим замовленням і не може бути безпосередньо (без будь-яких змін) використаний в інших системах. Крім того, висока трудомісткість і вартість ПТК робить економічно невиправданим їх знеособлений випуск («на склад»). Загальні вимоги до розробляється ПТК зазвичай встановлюють в ТЗ на ІУС. Вони випливають з проекту системи, компонентом якої є цей ПТК, і регламентують виконувані ним функції, основні характеристики, сигнали і інтерфейси з іншими компонентами, сполученими системами і обладнанням, які не залежать від способів реалізації ПТК.

Відповідно до ГОСТ 15.005 розробляється програмно-технічний комплекс, що розглядається як головний зразок, на базі якого згодом можуть виконуватися разові замовлення на виготовлення аналогічних ПТК (поставних комплектів), при цьому головний зразок зазвичай (хоча й не завжди) являється першим поставним комплектом. Згідно ГОСТ 15.005, документом, що встановлює вимоги до виробів одиничного виробництва і регулює відносини між постачальником (розробником, виробником) і споживачем (замовником) цих виробів, є технічне завдання.

У технічному завданні на розробку, виготовлення і поставку ПТК, як мінімум, регламентують вимоги:

до виконуваних основним і додаткових функцій;

надійності, стійкості, якості, незалежності виконання функцій;

програмному та інформаційному забезпеченню;

документам, які обгрунтовують безпеку;

управління якістю розробки та виготовлення;

оцінки та підтвердження відповідності на всіх етапах життєвого циклу.

Для кожної з виконуваних функцій в ТЗ встановлюють: цільове призначення; перелік контрольованих параметрів, подій і станів; діапазони зміни і межі

можливих значень контрольованих параметрів; види вхідних і вихідних сигналів; залежність між входами і виходами (у вигляді формул, таблиць або алгоритмів); інші дані, необхідні і достатні для реалізації цієї функції. У ТЗ зазвичай передбачають резерв обладнання та обчислювальної потужності ПТК, необхідний для того, щоб забезпечити можливість модернізації системи при зміні (розширенні) функціональних вимог на наступних етапах життєвого циклу, а також дтя компенсації виникаючих дефектів в ПТК, якщо заміна елемента, що відмовив тимчасово неможлива або недоцільна. Для всіх експлуатаційно-автономних складових частин ПТК регламентують

вимоги по стійкості або міцності до впливу зовнішніх факторів (навколишнього середовища, механічних, електричних, електромагнітних) у робочих і граничних умовах експлуатації. У ТЗ призводять також вказівки по експлуатації, в тому числі обсяг і періодичність технічного обслуговування, що використовуються при цьому методи, технічні засоби і програмне забезпечення (при необхідності); способи перевірки технічного стану ПТК, виявлення дефектів і відновлення; правила заміни або ремонту непрацюючих складових частин.

При розробці ТЗ на ПТК зазвичай беруть до уваги вихідні дані, підтримувані усіма передбачуваними замовниками поставних комплектів, щоб забезпечити відповідність кожної системи, яка в подальшому може бути реалізована на базі цього ПТК, вимогам, що пред'являються до неї. ТЗ на розробку, виготовлення і поставку ПТК, як вироби одиничного виробництва, погоджують із замовниками (експлуатуючою організацією) і органом державного регулювання ядерної безпеки.

В процесі розробки головного зразка ПТК визначають склад, структуру і розподіл функцій між його складовими частинами; оцінюють можливість і доцільність застосування апаратури, раніше розробленої для інших ПТК; приймають рішення про використання запозичених (тиражованих) і покупних (загальнопромислових) виробів; вибирають елементну і конструктивну базу для нових складових частин, які повинні бути розроблені для цього ПТК. Прийняті технічні рішення можуть коригуватися на наступних етапах з урахуванням технічної і економічної доцільності, а також на підставі оцінки їх відповідності

вимогам нормативних документів і ТЗ. Паралельно розробляється програмне забезпечення ПТК і проводиться верифікація на всіх етапах його створення - при формулюванні вихідних вимог, проектуванні, кодуванні, інтеграції з апаратними засобами. План і звіт по верифікації програмного забезпечення узгоджуються з органом державного регулювання ядерної безпеки. На завершальному етапі розробки випускається конструкторська і програмна документація, яка визначає склад і пристрій головного зразка і містить необхідні дані для його виготовлення, перевірки (контролю, випробувань) та приймання.

В процесі валідації виготовленого головного зразка підтверджують відповідність розробленого ПТК нормам і правилам ядерної та радіаційної безпеки і вимогам ТЗ. Валідацію проводять організація-розробник і виробник головного зразка відповідно до плану валідації, узгодженим з органом державного регулювання ядерної безпеки. Першим етапом валідації є попередні автономні випробування складових частин і комплексні випробування головного зразка з імітаторами джерел і приймачів сигналів. На другому етапі проводять приймальні випробування головного зразка за участю розробників, виробників, представників передбачуваних замовників і органу державного регулювання ядерної безпеки. Результати валідації, відбиті у відповідному звіті, акті і протоколах попередніх і приймальних випробувань головного зразка, направляються для контролю органу державного регулювання ядерної безпеки. Позитивні результати валідації служать підставою для відвантаження на АЕС головного зразка, якщо він є першим поставним комплектом, а також для розробки і виготовлення наступних поставних комплектів ПТК.

Повторне виготовлення одиничних виробів, можливість якого встановлена в ГОСТ 15.005, дозволяє не тільки істотно зменшити витрати на розробку і проектування ІКС, а й підвищити безпеку за рахунок використання апробованих системотехнічних, конструктивних, схемних і програмних рішень.

Стандарт передбачає, що після проведення випробувань і коригування документації головного зразка подальше виготовлення і поставка одиничних виробів може здійснюватися або на підставі розроблених технічних умов (ТУ), або

за тим самим чинним технічним завданням. По відношенню до ПТК слід мати на увазі, що випуск поставних комплектів по ТЗ не дасть виробнику можливість гнучко реагувати на вимоги замовників, оскільки будь-які зміни технічних умов вимагають узгодження з усіма організаціями, які раніше узгодили ці технічні умови, і реєстрації змін до порядку. При цьому всі зміни, внесені в ТУ і зареєстровані, поширюються на всі наступні поставні комплекти ПТК (в тому числі призначені для замовників, які ці зміни не ініціювали).

Другий варіант, передбачений ГОСТ15.005, - виготовлення та поставка тиражованих одиничних виробів за чинним ТЗ - видається більш простим, економічним і зручним як для постачальника, так і для замовників. Можливість мати єдине ТЗ для всіх ПТК (одного типу) обумовлена тим, що різниця між поставними комплектами і головним зразком, як правило, незначна, її порівняно легко врахувати при замовленні. Стандарт передбачає оформлення відмінностей замовленого одиничного виробу від головного зразка протоколом зміни ТЗ. Стосовно до ПТК, важливим для безпеки, такі зміни (доповнення) ТЗ:

враховують особливості проектованої системи, відбиті у вихідних даних і вимогах до розробки нових компонентів;

відносяться до постачань комплекту, призначеного для цієї системи;

розглядаються як невід'ємна частина загального ТЗ на ПТК при виконанні замовлення на розробку та виготовлення цього постачаємого комплекту.

затверджуються замовником (АЕС) і розробником (виробником, постачальником) ПТК;

узгоджуються з органом державного регулювання ядерної безпеки.

Тиражовані ПТК, орієнтовані на застосування в якості компонентів невизначеного безлічі ІКС, важливих для безпеки, розробляють і поставляють за правилами.

Згідно з визначенням, наведеним вище, до технічних засобів автоматизації (ТЗА) відносяться периферійні пристрої, допоміжне і сервісне обладнання ІКС та ПТК. Технічні засоби автоматизації- це одиниця продукції виробничо-технічного призначення, яка розробляється, поставляється і використовується в якості

експлуатаціонноавтономного компонента ІКС або ПТК. Таким чином, поняття ТЗА охоплює всі периферійні пристрої ІКС (датчики, що нормують перетворювачі, виконавчі механізми) і ПТК (відеомонітори, клавіатури, принтери), але не сам ПТК.

ТЗА мають схожість з ПТК, так як і ті. та інші представляють собою функціонально і конструктивно закінчені вироби, розроблені, виготовлені, пройшли заводські випробування. Виробник гарантує протягом зазначеного терміну відповідність постачаємих виробів вимогам, які наведені в конструкторської документації та документах на поставку. У той же час слід згадати наступні відмінності між ТЗА і ПТК.

ПТК ототожнюється з центральною частиною ІКС, в той час як ТЗА є периферійними пристроями, які забезпечують сполучення ІКС з технологічними системами і обладнанням і (або) з оперативним персоналом. Компонентами ІКС зазвичай є тільки один ПТК, функції якого тотожні функціям системи, і велика кількість ТЗА, кожне з яких виконує порівняно просту функцію в складі ІУС.

Як правило, ТЗА виконано у вигляді одного експлуатаційно-автономного виробу, місце розміщення якого, спосіб монтажу, умови експлуатації, електроживлення, заземлення, ступінь жорсткості електромагнітної обстановки однозначно визначені проектом, що дозволяє встановити клас безпеки, категорію сейсмостійкості, групи умов експлуатації та проживання, і регламентувати випливають з цієї класифікації вимоги, безпосередньо пов'язані з вашим ТЗА.

ПТК найчастіше представляє собою сукупність декількох експлуатаційно-автономних складових частин, які можуть мати різне значення для безпеки, експлуатуватися в неоднакових умовах і ставитися до різних класифікаційних груп, тому вимоги по стійкості до впливу зовнішніх факторів, завадостійкості, якості електричної ізоляції та інші можуть бути встановлені для кожної з цих складових частин окремо, але не для ПТК в цілому.

Експлуатаційно-автономні складові частини ПТК повинні з'єднуватися безпосередньо на місці експлуатації, що передбачає значний обсяг необхідних монтажних робіт за участю замовника, постачальника і, можливо, інших (спеціалізованих) організацій. ТЗА поставляються повністю зібраними на

підприємстві-виробнику, і при підготовці до експлуатації не вимагають проведення будь-яких додаткових складальних операцій, крім найпростіших.

Розрізняють одиничні ТЗА, які розробляються і поставляються на АЕС в якості компонентів конкретних ІКС або ПТК, важливих для безпеки; тиражовані ТЗА, спеціально розроблені, дозволені для застосування на АЕС і поставляються в якості компонентів заздалегідь не певного безлічі ІКС або ПТК, важливих для безпеки; загальнопромислові ТЗА, при розробці яких не ставилося завдання їх застосування в якості компонентів ІКС або ПТК, важливих для безпеки.

Розробка одиничних ТЗА передбачає, в основному, ті ж стадії, що і для ПТК Технічне завдання на розробку, виготовлення і поставку ТЗА має містити:

вимоги до виконуваних основним і додаткових функцій;

види і можливі діапазони зміни контрольованих фізичних величин, параметри вхідних і (або) вихідних сигналів, тимчасові і точності характеристики, а також інші дані, необхідні для реалізації запропонованих функцій;

робочі та граничні умови експлуатації (фактори навколишнього середовища, характеристики механічних і сейсмічних впливів, можливі зміни параметрів електроживлення, ознаки жорсткості електромагнітної обстановки), по відношенню до яких повинна забезпечуватися стійкість (стійкість або міцність) виробів;

вимоги до надійності, точності і тимчасовим характеристикам (при необхідності), розробки, виготовлення і постачання ТЗА, включаючи стратегію проведення випробувань;

вказівки з оцінки та підтвердження відповідності на наступних етапах життєвого циклу;

вимоги до програмного забезпечення, порядку його розробки та верифікації, якщо для виконання однієї або декількох основних або додаткових функцій передбачається використання програмованих електронних пристроїв (мікропроцесорів, однокристальних мікроЕОМ і т. п.).

У ТЗ повинні бути відображені всі застосовні вимоги норм і правил ядерно ної та радіаційної безпеки, що діють на момент розробки ТЗА. ТЗ на розробку,

виготовлення і поставку одиничного ТЗА узгоджують з передбачуваними замовниками (експлуатуючою організацією) і органом державного регулювання ядерної безпеки.

В процесі розробки головного зразка ТЗА вибирають основні технічні рішення (принцип дії, елементну базу, конструктивну будову і т. П.), які на наступних етапах можуть коригуватися за результатами оцінки відповідності вимогам нормативних документів і ТЗ. Для інтелектуальних ТЗА розробляється програмне забезпечення і проводиться його верифікація (план і звіт по верифікації узгоджуються з органом державного регулювання ядерної безпеки). На завершальних етапах розробки випускається конструкторська і програмна (лля інтелектуальних ТЗА) документація, виготовляється головний зразок (зразки) і проводиться валідація (попередні і приймальні випробування) для підтвердження відповідності розробленого ТСА нормам і правилам ядерної та радіаційної безпеки і вимогам ТЗ. План валідації, установлюючий обсяг, умови, методику перевірок (контролю, випробувань) та критерії відповідності головного зразка встановленим вимогам, узгоджується з органом державного регулювання ядерної безпеки, якому направляються також матеріали, що відображають результати валідації (звіт, акт і протоколи попередніх і приймальних випробувань). Позитивні результати валідації є підставою для відвантаження на АЕС першого (головного) зразка, а також для подальшого таражірованія ТЗА (виготовлення поставних зразків). акт і протоколи попередніх і приймальних випробувань). Позитивні результати валідації є підставою для відвантаження на АЕС першого (головного) зразка, а також для подальшого таражування ТЗА (виготовлення поставних зразків). акт і протоколи попередніх і приймальних випробувань). Позитивні результати валідації є підставою для відвантаження на АЕС першого (головного) зразка, а також для подальшого таражірованія ТЗА (виготовлення поставних зразків).

Повторне виготовлення і постачання одиничних ТЗА може проводитися за чинним ТЗ, при цьому допускаються деякі відмінності поставних зразків від головного, враховують особливості тієї системи, для якої вони призначені. Ці зміни (доповнення), що випливають з проекту системи (зафіксовані у вихідних даних і

вимогах до розробки нових компонентів - оформляються спільним протоколом замовника і постачальника, який узгоджується з органом державного регулювання ядерної безпеки і діє разом з ТЗ на всіх етапах життєвого циклу відповідних поставних зразків.

Всі поставні зразки ТЗА, призначених для застосування в системах, важливих для безпеки АЕС, підлягають спеціальному технічному прийманню та контролю (попередніми і приймально-здавальних випробувань) на майданчику виробника. Приймальний контроль проводиться з метою перевірки правильності виконання заданих функцій, визначення кількісних і якісних характеристик поставних зразків ТЗА і оцінки їх відповідності конструкторської документації, вимогам ТЗ і доповнень (змін) ТЗ. Акт і протоколи попередніх (заводських) та приймально-здавальних (з участю представників замовника) випробувань кожного зразка направляють для контролю органу державного регулювання ядерної безпеки. Позитивні результати приймального контролю дають підставу для відвантаження поставних зразків замовнику. Після транспортування, зберігання, монтажу та налагодження проводять передпускові випробування ТЗА, під час яких перевіряють і демонструють роботу поставлених зразків, їх відповідність вимогам ТЗ та додатковим змінам до ТЗ, готовність до інтеграції з іншими компонентами системи.

На відміну від ПТК, більш широке застосування в ІКС знаходять непоодинокі, а тиражовані ТЗА. Вимоги до таких виробів не можуть бути прив'язані до якої-небудь однієї системи: вони установлюються в ТЗ на розробку ТЗА (див. Рис. 2.2) і деталізуються в проекті технічних умов (ТУ), виходячи з власного уявлення розробника про призначення і можливі умови застосування цих виробів. Проект ТУ регламентує вимоги до функціонування, надійності, стійкості, якості, незалежності виконуваних функцій, правила приймання і методи випробування виробів. Відповідність розробленого ТЗА нормам і правилам ядерної та радіаційної безпеки і вимогам проекту ТУ підтверджують в процесі приймальних випробувань дослідного зразка (зразків),

Подальші етапи освоєння серійного виробництва відповідно до ГОСТ 15.001

передбачають: розробку нових технологічних процесів; виготовлення необхідного оснащення; випуск технологічної документації; виготовлення установчої серії (першої промислової партії) з використанням заводської технології, оснащення та обладнання; випробування зразків установчої серії; коригування, при необхідності, технічних умов, конструкторської та технологічної документації; привласнення документів літери «О». На підставі позитивних результатів, відображених в акті і протоколах випробувань установчої серії, орган державного регулювання ядерної безпеки приймає рішення про погодження ТУ, що дає можливість виготовляти і постачати на АЕС тиражовані вироби, предназначення для комплектації систем, важливих для безпеки. Кожен виріб, що поставляється, проходить спеціальне технічне приймання і піддається приймально-здавальним випробуванням службою контролю підприємства-виготовлювача на відповідність вимогам ТУ.

Розширення застосування загальнопромислових ТЗА в якості компонентів ІКС та ПТК, важливих для безпеки - одна з сучасних тенденцій, обумовлена масовим характером їх виробництва і гострою конкуренцією провідних світових лідерів в області інформаційних технологій, результатом чого яаляется істотне поліпшення споживчих властивостей (точності, швидкодії, завадостійкості та ін.), порівняно низька ціна, достатня надійність, апробованість в різних областях застосування, відсутність залежності від одного виробника, висока якість фірмового ервісного обслуговування . У той же час слід враховувати, що розробка, виготовлення і постачання загальнопромислових виробів виробляються без урахування норм та правил ядерної та радіаційної безпеки і не контролюються органом державного регулювання ядерної безпеки, а відомості про обсяг, методику та результати заводських випробувань практично недоступні для користувачів. Це обмежує застосування загальнопромислових виробів в системах, важливих для безпеки, де вони використовуються для виконання хоча й більш складних, але менш відповідальних функцій. Характерними прикладами є засоби, що підтримують інтерфейс «людина - машина», апаратура передачі даних по локальних мережах, промислові персональні ЕОМ і т. П. Можливість застосування таких виробів: пропонується для кожної системи за результатами їх кваліфікації.

Кваліфікація служить одним з методів зниження ймовірності відмов, викликаних зовнішніми подіями або впливом навколишнього середовища. Кваліфікація має підтвердити, що компоненти ІКС в змозі виконувати свої функції і зберігати необхідні характеристики протягом услановленого терміну служби в заданих робочих граничних умовах експлуатації. Кваліфікація передбачає: встановлення кваліфікаційних вимог; отримання відомостей про фактичні характеристики виробів; оцінку достовірності отриманих відомостей; порівняння створених характеристик з кваліфікаційними.

Для поодиноких ПТК і ТЗА, розроблених і виготовлених для застосування в конкретній системі, як кваліфікаційних приймають вимоги, установлені в ТЗ на розробку, виготовлення і поставку і в протоколі зміни ТЗ. Відомості про фактичні характеристики ПТК і ТСА отримують за результатами валідації, яка проводяться відповідно до правил, методами і критеріямі, регламентованими в плані валідації, узгодженому з органом державного регулювання ядерної безпеки. Оцінку достовірності отриманих відомостей і їх зіставлення з кваліфікаційними вимогами проводять в процесі незалежної експертизи ТЗ, протоколу зміни ТЗ, плану і звіту по валідації. Таким чином, в процесі проектування досить обмежитися визначенням вимог до розробки нових компонентів для системи, які слід встановлювати, керуючись діючими нормами і правилами ядерної та радіаційної безпеки, з урахуванням всіх очікуваних (передбачених проектом) умов в передбачуваних місцях розміщення виробів і необхідних «кваліфікаційних запасів». При цьому сама кваліфікація одиничних ПТК і ТЗА поєднується з етапами їх розробки і виготовлення і може бути проведена таким способом, який забезпечує максимальну достовірність отриманих результатів.

Для тиражованих ПТК і ТЗА кваліфікаційні вимоги установлюють аналогічним чином в проекті системи, в якій передбачається їх застосування. Відомості про фактичні характеристики отримують з ТУ на ці вироби (узгоджених з органом державного регулювання ядерної безпеки на підставі позитивних результатів випробувань установчої серії). Співставлення характеристик, регламентованих в ТУ, до кваліфікаційних вимог повинно проводитися в процесі

проектування системи. Виріб можна вважати кваліфікованим для застосування в даній системі, якщо вимоги, встановлені в ТУ, охоплюють весь перелік кваліфікаційних вимог та еквівалентні їм (або є більш жорсткими). Якщо окремі вимоги ТУ не збігаються (або не повністю збігаються) з кваліфікаційними, а також при відсутності в ТУ частини необхідних вимог, проводять додаткову перевірку (випробування) на відповідність кваліфікаційним вимогам - можливо, за участю розробника і (або) виробника тиражируемого вироби. Для подальшого застосування в аналогічних ІКС обсяг додаткових перевірок (випробувань) може бути обґрунтовано зменшений. Такий підхід дозволяє мінімізувати роботи по кваліфікації тиражованих ПТК і ТЗА без шкоди для безпеки.

Для загальнопромислових ТЗА процедура кваліфікації повинна додатково передбачати оцінку повноти та достовірності наявних відомостей про фактичні характеристики виробів і, в разі необхідності, їх перевірку за допомогою випробувань, аналізу, оцінки досвіду експлуатації або комбінації цих методів відповідно до ГОСТ.

За результатами кваліфікації тиражованих і загальнопромислових виробів, передбачуваних для застосування в проектованої системі, випускається звіт, в якому наводяться кваліфікаційні вимоги, метод проведення кваліфікації та отримані результати. Звіт повинен дозволяти компетентному персоналу, який не брав участі в кваліфікації, перевірити достовірність наведених відомостей і оцінити обґрунтованість зроблених висновків.

Оцінку правильності встановлення кваліфікаціонних вимог, достовірності відомостей про фактичні характеристики виробів, отриманих з документів постачальника і (або) за результатами випробувань, повноти і об’єктивні їх зіставлення з кваліфікаційними вимогами проводять в процесі незалежної експертизи звіту по кваліфікації, яку виконує організація, уповноважена органом державного регулювання ядерної безпеки.

Програмне забезпечення ІКС, ПТК, ТЗА - сукупність програм на зовнішніх носіях даних, в постійній і оперативної пам'яті системи або пристрої, а також відповідних програмних документів. У цьому визначенні під програмою

розуміється упорядкований набір команд і даних, які визначають дії у формі, придатній для виконання цифровими програмованими пристроями. Програмне забезпечення (ПЗ) розглядається в якості одного з компонентів ІКС починаючи з того часу, коли для виконання основних функцій системи стали використовуватися керуючі електронно-обчислювальні машини (ЕОМ) широкого призначення. На вітчизняних АЕС з реакторами ВВЕР-1000 такий підхід одним з перших був реалізований в блокової інформаційній системі «Комплекс-Титан 2», розробленої на базі керуючої ЕОМ сєвєродонецького СНВО «Імпульс» СМ-2М, яка випускалася серійно і орієнтувалася на застосування в автоматизованих системах, що створюються в різних галузях промисловості. У складі СМ-2М поставлялося системне ПЗ, яке забезпечувало роботу і обслуговування самої ЕОМ і установленних програм. Прикладне програмне забезпечення, безпосередньо пов'язане із завданнями контролю та упраатенія процесом, а не з функціонуванням самої ЕОМ, зазвичай розробляються незалежно від постачальника СМ-2М [для системи «комплекс-Титан 2» - Центратьним науково-дослідним інститутом комплексної автоматизації (ЦНДІКА)].

Системне програмне забезпечення прийнято розділяти на дві частини: операційне (програми, які безпосередньо виконуються в процесі роботи ЕОМ) і підтримує (програми, які використовуються при розробці, випробуваннях або обслуговуванні операційного програмного забезпечення та апаратної частини ЕОМ). Приклади операційного ПЗ: драйвери та сервіси введення-виведення; драйвери комунікації; програми управління переривань; програми планування завдань; програми оперативної діагностики, управління надмірністю і поступової деградацією при відмовах; бібліотеки прикладних програм. Приклади підтримує ПЗ: компілятори, генератори кодів, симулятори, програми автономного тестування, програмні утиліти і т. п.

Інтеграція розроблених прикладних програм з операційним програмним забезпеченням і апаратними засобами керуючої ЕОМ зазвичай здійснювалася на полігоні розробника прикладного ПЗ, а остаточне налагодження і перевірка системи - безпосередньо на місці експлуатації. Процес інтеграції та подальшої

перевірки (верифікації) відбувався в умовах гострого дефіциту часу, що ускладнювало виявлення помилок, які могли бути допущені при розробці прикладного ПЗ. Дефекти, внесені в процесі розробки і не виявлені при верифікації, могли в певних умовах проявлятися під час експлуатації і приводити до відмови функцій, які виконуються системою. «Приховані» дефекти ПЗ неможливо парирувати за рахунок резервування, якщо в резервованих частинах (каналах) системи реалізуються ідентичні програмні версії ПЗ.

Характерні риси описаного підходу до розробки програмного забезпечення ІКС (Рис. 2.2): орієнтація на єдину в системі керуючу ЕОМ; зберігання всіх виконуваних програм в пам'яті цієї ЕОМ; чіткий поділ ПЗ на системне (операційне підтримує) і прикладне; розробка пераційного та прикладного ПЗ різними організаціями. Недостатня надійність перших керівників ЕОМ широкого призначення, мале швидкодію, значна трудомісткість розробки і налагодження прикладного ПЗ, висока ймовірність «прихованих» дефектів, викликаних помилками програмування, і неможливість вичерпного тестування прикладного програмного забезпечення на полігоні розробника не дозволяли використовувати такі ЕОМ в системах безпеки.

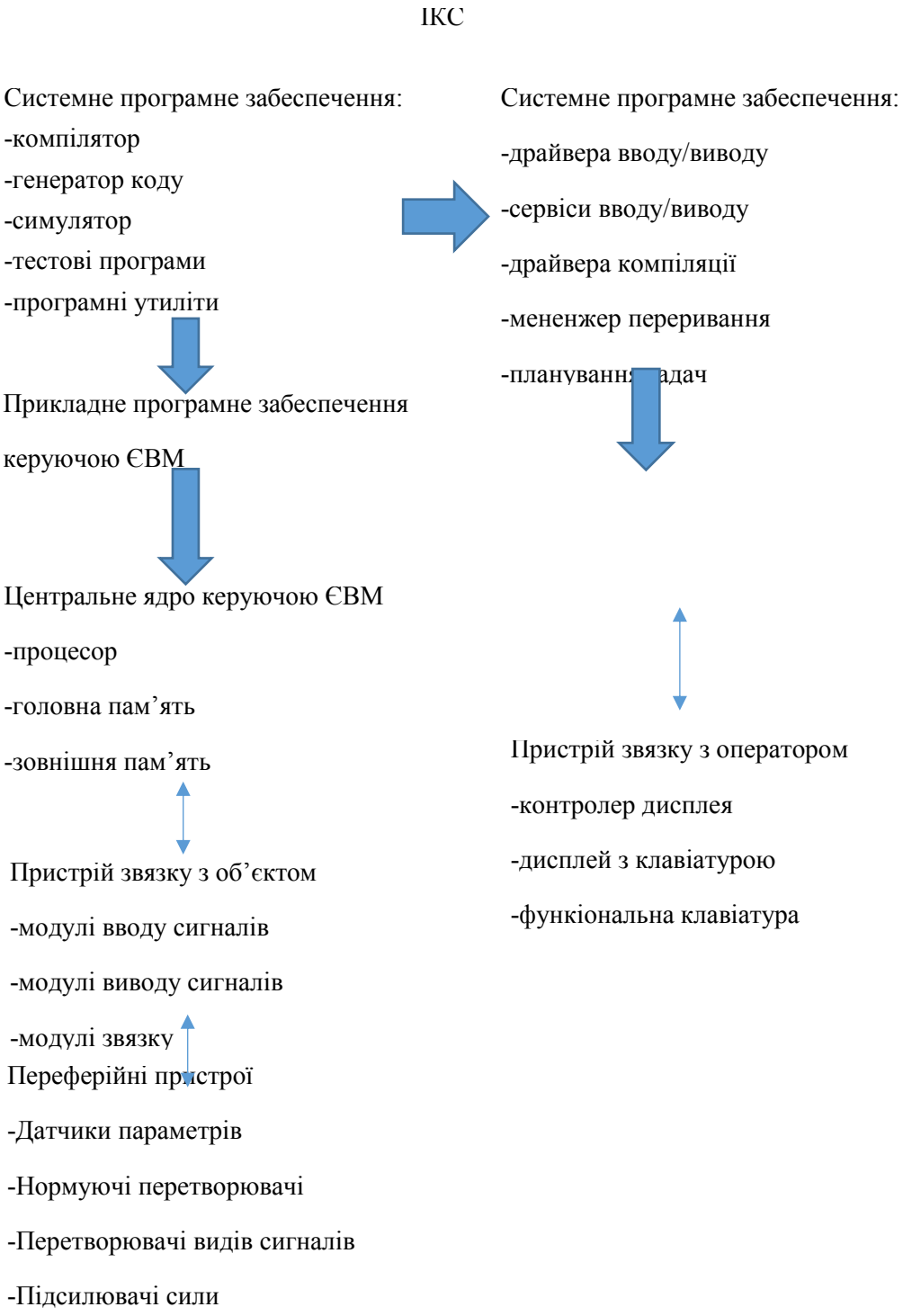


Рис. 2.2. Апаратні засоби та програмне забезпечення інформаційно керуючої системи

Істотне розширення складу програмно-реалізуємих або програмно-підтримуючих функцій на енергоблоках АЕС було забезпечено, в першу чергу,

розвитком елементної бази - появою доступних для використання мікропроцесорів, однокристальних мікроЕВМ і мікропроцесорних контролерів. включають один або декілька до мікропроцесорів разом з відповідною пам'яттю, для яких запропонована загальна назва programmable elektronic - програмовані електронні пристрої. У цьому ж стандарті система управління, захисту або моніторингу, яка базується на одному або декількох програмованих електронних пристроях, названа programmable electronic system - програмованої електронною системою.

З'явилася можливість здійснити функціональну декомпозицію - поділ складних системних функцій на ряд істотно більш простих, для реалізації кожної з яких було досить обчислювальних можливостей і обсягу пам'яті з'явилися в той час програмованих електронних пристроїв. Наприклад, могли бути виділені функції введення і перетворення вхідних сигналів, порівняння з уставками, логічної обробки і формування команд управління, безперервного контролю технічного стану і т. П. Поряд з функціональною, застосовувалася структурна декомпозиція, що передбачає участь у виконанні однієї функції декількох послідовно чи паралельно включених програмованих електронних пристроїв. Наприклад, при великій кількості вхідних сигналів функція введення і перетворення може здійснюватися не одним, а декількома одночасно працюючими пристроями, між якими розподіляються все вхідні сигнали; функція перевірки технічного стану може програмно підтримуватися безпосередньо в складових частинах системи, а результати перевірки можуть передаватися для подальшої обробки, відображення та архівування спеціально призначеним для цього програмованим електронним пристроям. Програмна підтримка всіх або частини виконуваних функцій може передбачатися також в окремих периферійних пристроях (інтелектуальних ТЗА) відображення та архівування спеціально призначеним для цього програмованим електронним пристроям. Програмна підтримка всіх або частини виконуваних функцій може передбачатися також в окремих периферійних пристроях (інтелектуальних ТЗА) відображення та архівування спеціально призначеним для цього програмованим електронним пристроям. Програмна підтримка всіх або частини виконуваних функцій може передбачатися також в окремих периферійних

пристроях (інтелектуальних ТЗА).

Такий підхід до створення програмованих електронних систем, що отримав назву децентралізованого, або розподіленого, управління, дозволив відмовитися від центральної керуючої ЕОМ, замінивши її безліччю програмованих електронних пристроїв, розосереджених по всій системі. Вважається, що кожен таке пристрій окремо володіє «малою складністю» (low complexity) в тому сенсі, що види відмов кожного з його компонентів чітко визначені, а поведінка пристрою в разі прояву будь-яких дефектів детерміновано. Простота і відносна незалежність виконуваних функцій кардинально спростили розробку ПЗ для програмованих електронних пристроїв малої складності і, зокрема, дозволили відмовитися від використання операційної системи, програмних переривань, драйверів і т. П. Важливим фактором являється також «видимість» програмного забезпечення таких пристроїв, що дозволяє зменшити ймовірність помилок при розробці ПЗ і порівняно легко виявляти дефекти в процесі тестування (до поставки замовнику). Таким чином, характерними рисами програмного забезпечення сучасних ІКС є: орієнтація ПЗ на багато програмовані електронні пристрої, що входять до складу системи; зберігання виконуваних програм безпосередньо в цих пристроях; стирання відмінностей між системним (операційним) і прикладним ПЗ; суміщення розробки апаратної частини і програмного забезпечення таких пристроїв, які виробляються одночасно однією і тією ж організацією.

Наслідком зазначених особливостей стала можливість програмної реалізації основних функцій в системах, важливих для безпеки АЕС. У другій половині 1980-х років на французьких реакторах PWR -1400 введені в лад програмовані електронні системи, розроблені фірмою Electricité de France (EDF), які виконували не тільки функції інформаційної підтримки персоналу, автоматичного регулювання, логічного управління, а й функції попереджувальної та аварійної захисту реактора. Аналогічні завдання вирішувалися Westinghouse Electric Company на англійській AEC Sizewell B, а також канадською корпорацією Atomic Energy of Canada Ltd. (AECL) в процесі удосконалення тяжёловодного реактора CANDU-3 на AEC Darlington-A.

У 2002-2003 роках на двох енергоблоках чешської АЕС «Темелін» (з реакторами ВВЕР-1000 російського виробництва) Westinghouse Electric Company впровадила автоматизовані системи управління технологічними процесами, в яких використані програмні методи реалізації функцій аварійного захисту, регулювання, обмеження потужності реактора, внутрішньо-реакторного контролю, управління технологічним обладнанням, відображення інформації на блочному і резервному щитах, а також функцій радіаційного контролю і післяаварійного моніторингу.

Настільки ж масштабно програмовані електронні пристрої застосовувалися в системах, які німецька Siemens Power Corporation (в даний час - Framatome ANP) реалізувала за останні 10 років на ряді європейських АЕС, а також на двох енергоблоках Тяньваньської АЕС в Китаї.

На радянських атомних енергоблоках програмовані електронні пристрої були вперше застосовані в системах: авто матичного управління турбінами ЛОТ-1000, розроблених на початку 1980-х голів харківським ВО «Моноліт» (в даний час - Харківський приладобудівний завод ім. Т. Г. Шевченка).

Зараз на більшості енергоблоків України всі основні функції системи управління і захисту реактора реалізуються програмно-технічними комплексами, основу яких складають програмовані електронні пристрої.

Для сучасного етапу розвитку інформаційних технологій характерне розширене використання одного з видів таких пристроїв, які в даний час прийнято називати складними програмованими електронними компонентами. Цим терміном позначають інтегральні мікросхеми, що настроюються на виконання заданих функцій шляхом конфігурації зв'язків між їх елементами. У більшості додатків вони успішно конкурують з мікропроцесорами загального призначення, оскільки працюють значно швидше за рахунок використання паралельного (а не послідовного, як в мікропроцесорах) способу реалізації алгоритмів. Складними програмованими електронними компонентами є, наприклад, замовні мікросхеми, що конфігуруються виробником згідно з вихідними даними, які представлені замовником, а також програмовані логічні інтегральні схеми (ПЛІС), що конфігуруються замовником (користувачем). Останні становлять найбільший

інтерес для розробників одиничних і тиражованих виробів.

Кожна ПЛІС виконана у вигляді поля з кількох сотень щодо великих логічних блоків (макроячейок). Макроячейки являють собою програмовану матрицю логічних елементів «І», а також елементів пам'яті, які можуть входити до її складу або формуватися з інших елементів. Макроячейки може містити кілька тисяч або десятків тисяч елементів. Алгоритм функціонування конкретної ПЛІС однозначно визначається сполуками логічних елементів «І» всередині кожної макроячейки, сполуками макроячейок одна з одною і з зовнішніми висновками мікросхеми. Елементи і зв'язки електронної схеми визначаються електронним проектом ПЛІС, який розробляється і реалізується користувачем. Для автоматизації проектування використовується спеціальна система розробки і верифікації електронних проектів, орієнтована на ПЛІС певного типу. Такі системи розробляють і випускають всі провідні виробники ПЛІС (Рис. 2.3).

Необхідні алгоритми функціонування ПЛІС задають:

на графічній мові (у вигляді схем алгоритмів, графіків вхідних та вихідних сигналів або електричної схеми в одному з прийнятих стандартних фор-матів), використовуючи бібліотеку примітивів в середовищі проектування (як запропонованих фірмою-постачальником, таки власної розробки);

із застосуванням стандартних мов опису апаратури (HDL - Hardware Description Languages) і (або) спеціатьних мов, розроблених фірмою поставшиков і враховують архітектурні особливості конкретних сімейств ПЛІС;

на одній з мов високого рівня - при використанні емуляторів стандартних мікропроцесорів, інтерфейсних контролерів, контролерів локальних мереж і т п. які імплементуються в логічну структуру ПЛІС в якості складових частин(«ядер» - irseikctual Property Cores), що реалізують типові мегафункції (для емуляції таких ядер використовується відповідна надбудова в середовищі проектування).

Рис. 2.3. Технологія розробки електронного проекту ПЛІС

Одна і та ж мікросхема може одночасно виконувати кілька функцій, при цьому атгоритми функціонування задаються для кожної з них незалежно. Способи завдання алгоритмів доповнюють один одного і можуть комбінуватися для випромінювання найкращих результатів. По заданому алгоритму автоматично визначається необхідна структура сполук логічних елементів і макроячейок ПЛІС, що беруть участь в його реалізації (програмна модель алгоритму в середовищі проектування). При емуляції стандартних ядер структура сполук визначається типом відповідного ядра (наприклад, мікропроцесора). На слідуючому етапі інтегруються всі програмні моделі алгоритмів, які має виконувати ПЛІС, і всі емулятори стандартних ядер (якщо вони використовуються). Результатом інтеграції являється програмна модель логічної структури ПЛІС і її представлення у вигляді файлу конфігурації. На завершальному етапі проектування ця програмна модель імплементується в ПЛІС шляхом автоматичної реалізації фізичних з'єднань її елементів з використанням технології та обладнання, які прийняті для відповідного сімейства ПЛІС. Кошти, необхідні для імплементації, входять до

складу системи розробки та верифікації електронних проектів, яка використовувалася при проектуванні логічної структури ПЛІС.

Завершення чергового етапу можливо тільки після верифікації продукту, створеного на цьому етапі (підтвердження його відповідності вимогам і вихідними даними, які були сформульовані на попередньому етапі). Кінцевим результатом проектування є ПЛІС з імплементувати логічною структурою, готова для установки в відповідний інтелектуальний блок (модуль).

Функціонування кожної ПЛІС в кінцевому підсумку визначається фізичними зв'язками між її елементами. В цьому відношенні вона принципово не відрізняється від пристроїв, які апаратно реалізують запропоновані функції (їх називають також пристроями з жорсткою логікою). У той же час сама технологія розробки електронних проектів ПЛІС має багато спільного з розробкою прикладного ПЗ для програмованих електронних пристроїв (використання спеціальних систем автоматизації програмування, відповідних інструментальних засобів, мов програмування високого рівня, процедур поетапної верифікації і т. п.). З цієї причини розробку електронних проектів ПЛІС прийнято розглядати як один з видів програмування, хоча і дещо відрізняється щодо вимог до розробки і верифікації. Якщо ж в ПЛІС емулюватися стандартні мікропроцесори, прикладне ПЗ для них розробляється і перевіряється за тими ж правилами, що і для інших програмованих електронних пристроїв.

Основні проблеми при впровадженні програмованих електронних систем, важливих для безпеки, з самого початку були пов'язані з забезпеченням і доказом надійності ПЗ. Складність полягає в тому, що для оцінки надійності технічних засобів і програмного забезпечення необхідні принципово різні методи, оскільки при аналізі надійності апаратури постулюється випадкові відмови, в той час як відмови ПЗ викликані, головним чином, помилками розробників, що приводять до появи систематичних дефектів(іншими причинами відмов, часто не зовсім коректно відносяться до відмов ПЗ, можуть бути вплив комп'ютерних вірусів і відмови апаратних засобів, в яких зберігаються константи або команди ПЗ). Діючі в Україні стандарти, наприклад , пов'язують надійність зі здатністю виконувати

необхідні функції; за кордоном цю властивість іноді визначають як готовність, характеризуючи його безвідмовністю і ремонтопридатністю.

Однак готовність виконувати необхідні функції ще не дає впевненості в правильності виконання цих функцій. Тому в міжнародному стандарті додатково вводять в розгляд властивість достовірності. Стосовно до програмного забезпечення воно вимагає, щоб ПЗ було здатне:

саме розпізнавати і попереджати про перехід в такий стан, при якому деякі функції можуть не виконуватись належним (безпомилковість);

запобігати несанкціонований доступ і (або) некоректна зміна програм і даних (захищеність).

Забезпечення достовірності особливо актуальне для програмованих електронних систем, важливих для безпеки. Виходячи з цього в останніх наукових публікаціяхі нових міжнародних стандартах, що відносятся до таких систем та їх програмного забезпечення, прийняте більш широке трактування такої властивості, яка традиційно позначається терміном depen¬dability: тепер вона включає не лише готовність виконувати певні функції, але й достовірність та захищеність. З урахуванням такого розширення було запропоновано нові еквіваленти терміну dependability — надійність або гарантоспроможність.Сучасні технології розробки гарантоздатного ПЗ для програмованих електронних пристроїв і систем, важливих для безпеки, а також норми і правила оцінки такого ПЗ наведені в міжнародних стандартах. Зокрема, вказується, що програмне забезпечення, критичне для безпеки, має бути обмежене абсолютно необхідним мінімумом і відокремлене від решти ПЗ. Рекомендується також чіткий поділ між прикладним, системним і сервісним програмним забезпеченням. Проект ПЗ повинен бути добре структурований і мати чітко визначені і, по можливості, мінімальні зв'язку між компонентами. Статична диспетчеризація виконуваних операцій краще, ніж використання переривань, оскільки дозволяє виявляти відхилення від детермінованого поведінки і автоматично відновлювати нормальну роботу програми при збоях, тимчасової втрати вхідної інформації іт.п., наприклад за допомогою сторожового таймера.

Гарантоздатність досягається не тільки дотриманням вимог до створюваного програмного продукту, а й виконанням рекомендованих правил розробки, перевірки і модифікації програмного забезпечення. Наприклад, вже на початку проектування слід розробити суворі, однозначно зрозумілі специфікації архітектури програмного забезпечення, функцій системного ПЗ, прикладних і сервісних функцій, які повинні виконуватися відповідними програмам. практика». Характерною особливістю сучасного підходу до розробки ПЗ являється використання інструментальних засобів автоматизації розробки, налагодження, тестування та модефікації програмного забезпечення.

В національних та міжнародних стандартах регламентовані вимоги і критерії, що дозволяють оцінити можливість використання раніше розроблених програмних продуктів, включаючи COTS, як інструментальних засобів розробки, тестування та модифікації ПЗ, критичного для безпеки, і (або) безпосередньо в складі такого ПЗ.

Для виявлення дефектів, викликаних помилками розробників, передбачена верифікація ПЗ (перевірка і підтвердження узгодженості результатів, отриманих на кожному етапі розробки, з вимогами, встановленими на попередніх етапах, і надання об'єктивних доказів того, що ці вимоги були виконані). Відповідно до національногостандарту, верифікація ПЗ повинна проводитися після завершення кожного етапу (розробки специфікацій, проектування, кодування), охоплювати не тільки отриманий продукт, а й сам процес його розробки, включати аналіз і усунення виявлених та зафіксованих недоліків. На завершальних етапах, після об'єднання складових частин і інтеграції ПЗ з апаратними засобами, розроблене ПЗ піддається множинним перевіркам із застосуванням (там, де це можливо) автоматичного тестування.

Фахівцями різних галузей промисловості, в тому числі атомної енергетики, широко використовується термін «автоматизована система управління технологічним процесом» (АСУ ТП). Термінологічний стандарт визначає АСУ ТП як систему, що складається з персоналу і комплексу засобів автоматизації його діяльності, яка реалізує інформаційну технологію виконання встановлених функцій управління технологічним процесом.

Тим часом на практиці під АСУ ТП розуміють за замовчуванням лише одну її частину - комплекс засобів автоматизації: тільки в такому аспекті можна трактувати, наприклад, вимоги нормативних документів до надійності, безпеки, ергономіки, проведення випробувань, комплектності АСУТП, гарантіями та ін. В нормах і правилах ядерної та радіаційної безпеки, що діють в Україні та Росії, поняття АСУ ТП не використовується, проте в новому документі введений близький за змістом термін «автоматизована система контролю і управління технологічними процесами енергоблоку», яка «повинна забезпечити дистанційне і (або) автоматичне керування технологічними процесами і системами безпеки, автоматичний захист систем, обладнання та енергоблоку в цілому, а також контроль за неперевищенням меж безпечної експлуатації енергоблоку ».

Хоча застосування в назві слова «автоматизована», здавалося б, підкреслює участь персоналу в роботі системи, однак аналіз наведених вимог до її складу і виконуваних функцій дозволяє зробити висновок про те, що персонал енергоблоку лише взаємодіє з системою, але в її склад не входить. Залишається відкритим також питання про місце керуючих систем безпеки: вони безпосередньо беруть участь у виконанні таких функцій, як контроль за неперевищенням меж безпечної експлуатації, автоматичне керування технологічними системами безпеки, автоматичний захист обладнання, однак розглядаються окремо від системи контролю і упрааленія технологічними процесами енергоблоку.

У міжнародних стандартах відсутнє поняття, аналогічне АСУТП.

Під терміном «Instrumentation and control system» розуміють, в залежності від контексту, як окрему інформаційну або керуючу систему (наприклад, систему аварійного захисту або блокову інформаційно-обчислювальну систему), так і сукупність таких систем, а в деяких випадках - програмно-технічний комплекс. Іноді для того, щоб позначити саме сукупність систем, використовують прикметник overall (загальна), наприклад в довіднику «overall process control system» або «overall plant control». Інший підхід прийнятий в стандарт: тут окрема система позначається словом individual (індивідуальна або одинична).

Щоб зняти зазначені суперечності, був запропонований термін «система управління технологічним процесом», під якою розуміється сукупність всіх інформаційних і керуючих систем енергоблоку. Це поняття охоплює як системи нормальної експлуатації, так і системи безпеки, але не включає персонал енергоблоку. У наступних розділах воно буде використано, при необхідності, в тому ж самому сенсі, в якому термін АСУ ТП застосовується в міждержавних стандартах, що відносяться до інформаційних і керуючих систем атомних станцій.

можливість реалізації

На Рис. 2.4 показані відносини між поняттями, обговорюваними в цьому розділі.

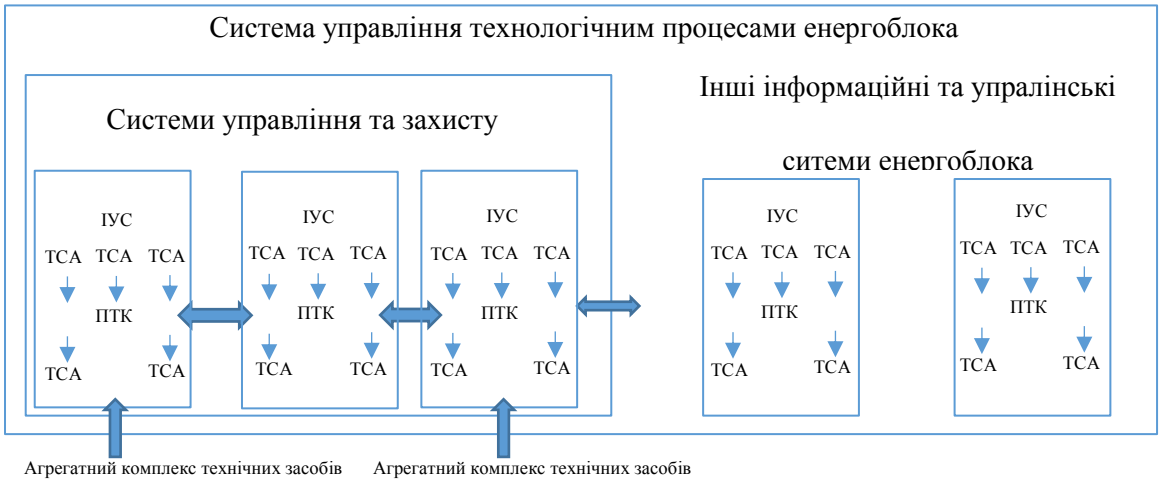


Рис. 2.4. Системи та компоненти автоматизованої системи управління

Граничними елементами АСУ ТП з боку входів є датчики та інші засоби отримання інформації про контрольовані і керованих фізичних величинах (параметрах), події і стани технологічних систем і устаткування, а також засоби ручного введення даних, команд і директив персоналу. Граничні елементи з боку виходів - виконавчі пристрої, що безпосередньо впливають на технологічні системи й устаткування (виконавчі механізми, потужні підсилювачі, контактори, магнітні пускачі і т. П.), а також засоби відображення і реєстрації даних. До складу системи не включають кабельні гермопроходки і елементи технологічних систем і

устаткування, призначені для зміни їх стану, властивостей або функціонування (наприклад, трубопровідну арматуру, що поглинають стрижні і ін.).

У визначенні відображена ситуація, реально склалася на українських АЕС: фактично замість єдиної системи управління технологічними процесами на кожному енергоблоці використовується своя, історично сформована сукупність окремих інформаційних і керуючих систем, створених і (або) модернізованих в різний час і на різних платформах.

За кордоном налічується порівняно небагато АЕС, на яких реалізовані основні системні принципи, характерні для систем управління технологічними процесами енергоблоків: спадність проектування; оптимальний розподіл функцій між коспонентами підсистемами;

обґрунтоване обмеження при виборі платформ для цих підсистем;

уніфікація канатів, інтерфейсів і протоколів передачі даних;

загальна база даних, загальне програмне, інформаційне та метрологічне забезпечення.

Такі результати були досягнуті, наприклад, Westinghouse Electric Company при модернізації проектних систем на енергоблоках АЕС Temelin в Чехії. Однак необхідність капітальної переробки проекту, заміни великої кількості діючих систем контролю і управління, сполученого устаткування і кабелів, а також значний обсяг пусконалагоджувальних робіт і випробувань безпосередньо на майданчику АЕС привели до суттєвої затримки пуску цих енергоблоків.

Доречно відмітити, що галузь ядерного приладобудування України також має в розпорядженні можливості для створення системи упрааления технологічними процесами енергоблока, яка могла б мати усі властивості, необхідні для АСУТП згідно ГОСТ 24.104 і іншим документам Єдиної системи стандартів автоматизованих систем управління, і відповідати чинним нормам і правилам ядерної і радіаційної безпеки. Вітчизняні організації мають певний досвід в розробці і впровадженні подібних систем. Зокрема, ТОВ "Вестрон" розроблені (на платформах "Вулкан" і "Вулкан-м"), виготовлені, поставлені на енергоблоки №№ 3 і 4 теплових електростанції "Південний Багдад" і запущені програмно-технічні

комплекси, які виконують усі функції АСУ ТП, включаючи захисту, блокування, автоматичне управління і регулювання технологічних процесів, ручне управління безпосередньо з екранів робочих станцій БЩУ, а також функції верхнього блокового рівня (енергоблок № 4 підключений до енергосистеми Іраку в грудні 2008 року, енергоблок №3 – в червні 2009)

Агрегатні комплекси технічних засобів МСКУ і ПС 5120, розроблені СНПО "Імпульс", також мають усе необхідне для збору і обробки даних про технологічні параметри і стан устаткування, автоматичного регулювання і управління, реалізації технологічних защит і блокувань, дистанційного керування, сигналізації, візуалізації на відеомоніторах і реєстрації процесу. У НПП "Радій" накопичений досвід координації робіт по проектуванню, монтажу, наладці і випробуванням на АЕС складних систем, що управляють, важливих для безпеки, і є необхідні засоби, що дозволяють в стислі терміни виготовляти і постадлять для цих систем ПТК високої заводської готовності. У усіх організаціях реалізується система технічних і організаційних заходів по забезпеченню якості, які охоплюють компонування, виготовлення і випробування ПТК, авторський нагляд за проведенням монтажно-налагоджувальних робіт і випробувань, а також гарантійне обслуговування поставленої продукції. Останнім часом роботи по модернізації інформаційних систем, що управляють, на українських АЕС базуються виключно на вітчизняних платформах. Проте, помітимо, що повномасштабні системи управління технологічними процесами можуть створюватися тільки на нових енергоблоках АЕС.

Висновки

Охарактеризовано основні функції інформаційно керуючої системи (ІКС), процеси життєвого циклу програмного забезпечення програмно-технічних комплексів критичного призначення.

Кожна ІКС призначена для конкретного технологічного об'єкта і, як правило, не може бути повторена без будь-яких змін для інших об'єктів. Це обумовлено:

відмінностями в технології і обладнанні об'єктів контролю і управління,

наприклад для енергоблоків українських АЕС - типом ядерного реактора (ВВЕР-440, ВВЕР-1000) і проектом ядерної установки (В-213, В-302, В-338, В-320);

відмінностями, що склалися в результаті попередніх модернізацій суміжних інформаційних і керуючих систем, з якими повинна взаємодіяти ІКС;

необхідністю заміни деякої частини наявних периферійних пристроїв (датчиків, виконавчих механізмів, рідше - сполучних кабелів), які фізично і морально застаріли;

постановкою нових завдань, установкою більш високих вимог до властивостей системи, що випливають із загальної тенденції розвитку інформаційних технологій, результатів науково-технічних досліджень і розробок, досвід експлуатації аналогів, нових нормативно-правових актів, міжнародних стандартів і т. п.

2.2. Процеси життєвого циклу програмного забезпечення програмно-технічних комплексів критичного призначення

Важливість проблеми безпеки атомних електростанцій важко переоцінити. Після Чорнобильської аварії значно підвищилися вимоги до питання експлуатації АЕС. А звідси і підвищені вимоги до роботи автоматики, або, іншими словами, до всіх інформаційних і керуючих систем, які створюють умови, що попереджають порушення нормальної експлуатації АЕС та значно зменшують наслідки аварії.

Необхідність модернізації

Прийняті в типовому проекті системні принципи побудови, склад і загальна структура АСУ ТП, а також архітектура інформаційних і керуючих систем і розроблене для них устаткування ґрунтувалися на можливостях радянської промисловості першої половини 1980-х років, однак уже тоді вони не відповідали рівню розвитку закордонної техніки. До того ж нормативні документи СРСР, використані при проектуванні й розробці, не відбивали всіх вимог, що діяли в той час міжнародних стандартів ядерної й радіаційної безпеки. Тому після реалізації типового проекту на перших енергоблоках з реакторною установкою моделі В-320

передбачалося в якості наступного кроку створення «перспективної АСУ ТП» для атомних енергоблоків, що будуються й проектованих у СРСР. Її основи закладалися фахівцями Інституту проблем керування Академії наук СРСР (ІАТ), Інституту атомної енергії ім. І. В. Курчатова, ЦНДІКА, Всесоюзного науково-дослідного інституту атомних електростанцій (ВНДІАЕС), інституту «Атомтеплопроекл», ВНДІЕМ, СНДІП, СНПО «Імпульс» та інших організацій. Були розроблені й затверджені «Основні технічні вимоги до АСУ ТП АЕС із реакторами ВВЭР-1000», випущене технічне завдання на створення перспективної системи, почалася розробка проекту. Передбачалося застосування загальних для всієї АСУ ТП системних принципів, архітектурних розв'язків і інтерфейсів; максимально можлива уніфікація елементо-конструктивної бази, технічних засобів і програмного забезпечення; широке використання більших інтегральних схем і мікропроцесорів вітчизняного виробництва .

У цей ж час були переглянуті діючі в СРСР і розроблені нові нормативні документи, у яких були сформмовані загальні положення безпеки атомних станцій, правила ядерної безпеки реакторних установок, вимоги до систем керування технологічними процесами АЕС і їх компонентам. Одночасно проводилася стандартизація технічних вимог до приладів, засобів автоматизації й систем керування загальнопромислового призначення та інш.). Передбачалося, що нові стандарти будуть ураховуватися при проектуванні й розробці систем :•; компонентів для перспективної АСУ ТП.

Однак після 1991 р. ці роботи припинилися. Аналогічна спроба розробки системи керування технологичними процесами для нових атомних енергоблоків, що проводилась в Україні з 1993-1995 рр., також не була завершена у зв'язку з оголошенням мораторію на їхнє будівництво. У той же час продовжити експлуатацію діючих АЕС і реалізувати розв'язок про добудування блоків № 4 Рівненської АЕС і № 2 Хмельницької АЕС, прийняте після скасування мораторію, було б неможливе без проведення модернізації діючого устаткування, особливо в умовах твердого контролю безпеки атомних електростанцій України з боку міжнародних організацій. У першу чергу, модернізація повинна була торкнутися

систем і компонента АСУ ТП як найбільш динамічної, гнучкої й важливої для безпеки часта загального проекту енергоблоку.

Мета та організація робіт.

Головною метою модернізації повинне було стати усунення *дефіцитів, що були,* безпеки проектної АСУ ТП, до яких відосяться, зокрема:

низька надійність застосованих технічних засобів;

недостатня глибина діагностування систем і компонентів;

незадовільна якість інтерфейсу «людей — машина»;

відсутність в оперативного персоналу чітко структурованих даних, достатніх для оцінки стану безпеки енергоблоку під час експлуатації й при аваріях;

невідповідність вимогам нормативних документів у частині сейсмостійкості, завадостійкості, пожежобезпечності та ін.;

непропорційні апаратурні витрати через малий ступінь інтеграції застосованої елементної бази, що суттєво ускладнювало технічне обслуговування.

Актуальність модернізації обумовлювалася також і тим, що за час, що пройшов після реалізації типового проекту, ще більш збільшився розрив між технічним рівнем систем і компонентів, експлуатованих на АЕС України, і можливостями, які відкриває застосування нових інформаційних технологій, електронних компонентів, засобів обчислювальної техніки й сполученого устаткування передачі, відображення, реєстрації даних і т.д. Крім того, фактичний строк експлуатації багатьох виробів перевищив показники довговічності, регламентовані в їхній технічній документації, наслідком чого стало не тільки моральне, але й фізичне старіння цих виробів. Паралельно практично припинилася підтримка з боку виготовлювачів таких технічних засобів, які перестали випускати аналогічну продукцію, змінили профіль роботи, були реорганізовані або ліквідовані. Це ж відноситься й до постачальників запасних частин для цих виробів.

Модернізація проектних АСУ ТП на енергоблоках АЕС України проводилася поетапно, відповідно до затвердженої Національної енергетичної компанії (НАЭК) «Енергоатом» «Програмою вузлової заміни підсистем енергоблоків ВВЕР-1000 і ВВЕР-440 на 2000-2006 роки» (була потім продовжено на 2007-2010 роки).

Оскільки під час планово-запобіжного ремонту діючого енергоблоку повна заміна всієї апаратури не може бути забезпечена ні економічно, ні організаційно, модернізація не могла торкатися загальної структури й складу проектної АСУТП: зміни стосувалися її окремих складових частин (інформаційних і керуючих систем), які модернізувалися по черзі й незалежно друг від друга. Поряд із цим створювалися й нові інформаційні системи, не передбачені типовим проектом. До них відносились, наприклад, системи підбору параметрів безпеки СППБ , впроваджені на 11 українських енергоблоках з реакторами ВВЕР-1000. Були відсутні в типовому проекті й системи інформаційної підтримки, які могли б забезпечити персонал АЕС і сторонніх експертів по безпеці інформацією про виникнення аварійної ситуації, розвитку аварії, стану й функціонуванні систем та устаткування енергоблоку під час аварії й у післяаварійний період. Такі системи, необхідні для керування проектними й запроектними аваріями й ліквідації їх наслідків, розроблені СНПО «Імпульс» (перша черга системи, реалізованої за принципом «чорного ящика», уведена в експлуатацію на енергоблоках № 1 і № 2 Хмельницької АЕС і енергоблоках № 3 і № 4 Рівненської АЕС).

Нормативною базою модернізації стали « Загальні положення забезпечення безпеки атомних станцій» , перероблені й уведені в дію в 2000 р., а також норми й правила, що безпосередньо відносяться до інформаційних і керуючих систем АЕС, у яких враховані «Основні технічні вимоги до АСУТП АЕС із реакторами ВВЕР-1000» і рекомендації, що діяли в той час в межах міжнародних стандартів та інструкцій по безпеці атомних електростанцій тієї частини, яка не суперечила законодавству й нормативній базі України.

Основні принципи. Для усунення дефіцитів безпеки, які були в проектній АСУ ТП. і забезпечення функціональної безпеки нових і модернізованих І ВУС були запропоновані й реалізовувалися наступні принципи.

Кожна ІКС призначена для конкретного енергоблоку й повинна була враховувати вимоги замовника (АЕС) технології, що випливають із особливостей технологічних систем і устаткування, зв'язків з іншими ІКС за станом на момент впровадження, особливостей роботи оперативного персоналу й т.п.

2 При модернізації ІКС зберігалися наскільки це можливо, апробовані технологічні алгоритми керування, які були реалізовані на діючих енергоблоках у відповідності з проектом.

3. Усі нові й модернізовані системи повинні були відповідати нормам та правилам ядерної й радіаційної безпеки, що діють в Україні.

4. До розробки, виготовлення й поставки компонентів ІКС залучались, наскільки це було можливо, організації й підприємства України для того, щоб полегшити взаємодію між замовником і виконавцями, спростити проведення авторського нагляду, обслуговування й супроводу в гарантійний період, а також наступне поповнення комплекту запасних частин і т.п.

5. У більшості випадків модернізація передбачала заміну всієї центральної частини діючої системи новим програмно-технічним комплексом, у той час як периферійний пристрої звичайно або зберігалися, або замінялися частково (в основному, у зв'язку з фізичним зношуванням) і, як правило, незалежно від ПТК.

6. Програмно-технічні комплекси для нових і модернізованих систем розроблялися й виготовлялися по разових замовленнях як виробу одиничного виробництва, що збираються на місці експлуатації, і поставлялися з максимально високою заводською готовністю — налагодженими, перевіреними, укомплектованими необхідними засобами монтажу, запасними частинами, сервісним устаткуванням, програмною й експлуатаційною документацією.

7. Кожний програмно-технічний комплекс, важливий для безпеки, проходив у встановленому порядку всі передбачені процедури узгодження з органом державного регулювання ядерної безпеки й мав дозвіл на застосування в конкретній системі.

8. Для заміни периферійного устаткування переважно використовувалися тиражовані ТЗА, дозволені для застосування на АЕС і кваліфіковані з обліком їх призначення, ролі в забезпеченні безпеки й конкретних умов експлуатації в модернізуємій системі. Не виключалася можливість застосування належним чином кваліфікованих загальнопромислових виробів, як це прийняте у світовій практиці.

Відповідно до вимог, нові й модернізовані системи і їх компоненти повинні

були відповідати досягнутому рівню науки, техніки й технології, який підтверджений науковими дослідженнями й практичним досвідом. Стосовно до ІКС, ПТК і ТЗА це передбачає:

перехід від аналогових до цифрових методів обробки, обміну даними й управління з використанням розподілених структур, локальних мереж і волоконно-оптичних ліній передачі цифрових повідомлень;

заміну окремих аналогових і цифрових приладів кольоровими відеотерміналами з поліпшеними споживчими властивостями, застосування способів вистави відображуваної інформації, найбільш зручних для сприйняття й аналізу;

широке використання сучасної апаратури ручного введення команд, відображення й реєстрації даних, програмних методів і мікропроцесорних засобів підтримки інтерфейсу «людина— машина»;

наявність у кожному ПТК вбудованої діагностичної системи, що здійснює: контроль стану власних апаратних і програмних засобів, сполученого устаткування й ліній передачі даних; виявлення відмов і несправностей на мінімально можливому рівні; відображення й реєстрацію діагностичних повідомлень;

архівування й довгочасне зберігання в необхідному обсязі поточної інформації й діагностичних повідомлень із такою розв'язною здатністю за часом, яка дозволяла б чітко розрізняти порядок проходження подій і зміни станів у будь-яких перехідних режимах;

переважне використання комплектуючих виробів провідних світових постачальників, у достатньому ступені апробованих, що й мають кращі характеристики серед інших аналогічних виробів.

Виконання робіт

У проведенні модернізації — розробці, виготовленні, випробуваннях і поставці устаткування й (або) розробці програмного забезпечення ПТК — брали участь зазначені вище й інші вітчизняні й закордонні організації й підприємства.

Модернізація проектної АСУТП для нових енергоблоків № 2 Хмельницької АЕС і № 4 Рівненської АЕС мала деякі особливості, оскільки почалася ще до

їхнього пуску. Після оголошення мораторію на обидві АЕС були вжиті необхідні заходи по збереженню раніше поставленої апаратури. Вироби, які перебували в приміщеннях, що відповідали вимогам нормативних документів до умов їх зберігання, консервувалися без демонтажу; а якщо ні, то після демонтажу й консервації вони переміщалися на зберігання в підходящі приміщення. Умови зберігання періодично контролювалися, проводилися переконсервація й вибіркова перевірка виробів за допомогою стендового устаткування. Після рішення про добудову блоків виконані обстеження технічного стану й перевірка устаткування, що підлягало заміні, по програмах, погоджених з Державним комітетом ядерного регулювання України.

При модернізації проектної АСУ ТП нових енергоблоків замінені:

-центральна частина системи аварійного й попереджувального захисту — двома комплектами ПТКАЗ-ПЗ (розроблювач і виготовлювач — НВП «Радій»);

-автоматичний регулятор потужності реактора АРМ5 і пристрій розвантаження й обмеження потужності РОМ2 — комплектом ПТКАРМ-РОМ-УПЗ [2.66] (розроблювач і виготовлювач — НВП «Радій»);

-система групового й індивідуального керування СДІУ разом із кроковими електромагнітними приводами — системою СДІУ-М і електромагнітними приводами ШЕМ ческого виробництва (розроблювач і виготовлювач — ШКОДА);

-апаратура контролю нейтронного потоку АКНП-З — апаратурою АКНП-І: двома комплектами, що забезпечують одержання інформації *для* СУЗ, операторів БЩУ й контролю перевантаження ядерного палива, і одним комплектом *для* підтримки персоналу, що здійснює керування із РЩУ (розроблювач і виготовлювач — СНПО «Імпульс»);

-система внутрішньореакторного контролю СВРК-01 енергоблоку №2 Хмельницької АЕС — системою СВРК-М із програмним забезпеченням « Хортиця-М» [2.68] (розроблювач і виготовлювач базового ПТК на платформі МСКУ-2 — СНПО «Імпульс», розроблювачі алгоритмів і прикладного програмного забезпечення — РНЦ «Курчатовський інститут», ІНІТ, ХІКА);

-система внутрішньореакторного контролю СВРК-01 енергоблоку № 4 РАЕС — системою СВРК-М із програмним забезпеченням «Круїз» (розроблювач алгоритмів і прикладного програмного забезпечення — ЗАТ «СНДІП- АТОМ»);

-автоматичні регулятори реакторного відділення — програмно-технічними комплексами систем нормальної експлуатації реакторного відділення ПТК САР СНЕ РО (розроблювач і виготовлювач — завод ім. Т. М. Шевченка, розроблювач алгоритмічної частини — ВАТ «Львіворгрес») і ПТК СНЕРО (розроблювач і виготовлювач — СНПО «Імпульс»);

-автоматизована система керування турбіною АСУТ-1000-2 і автоматичні регулятори турбінного відділення — програмно-технічними комплексами систем нормальної експлуатації турбінного відділення (включають також електричну частину електрогідравлічної системи регулювання турбіни) ПТК САР СНЕ ТЕ й ЭГСР (завод ім. Т. М. Шевченко й ВАТ «Львоворгрес») і ПТК СНЕ ТО | СНПО «Імпульс»);

-керуюча обчислювальна система « Комплекс-Титан 2» і установка централізованого контролю турбогенераторів А-701 — системою « Комплекс-АЕС» з робочими місцями РМОТ-ОЗ | розроблювач і виготовлювач базового ПТК на платформах МСКУ-2 і ПС5120 — СНПО «Імпульс», розроблювач алгоритмів і прикладного програмного забезпечення — ХІКА);

-пристрій керування перевантаженням ядерного палива — системою керування машиною перевантажувальної СУМП (розроблювач і виготовлювач механічної частини — Ganz Energetics Machinery , електронної частини — ТОВ «Evig»).

Виготовленню й поставці нових компонентів на Хмельницьку й Рівненську АЕС передувала їхня апробація на інших українських і закордонних атомних станціях.

Крім зазначених компонентів у процесі модернізації проектної АСУ ТП на енергоблоках № 2 Хмельницької АЕС і № 4 Рівненської АЕС замінялися:

- блоки (в обґрунтованих випадках — і шафи) уніфікованого комплексу технічних засобів УКТС — на нові блоки й шафи УКТС ДПІ, УКТС ВЛ і УКТС-Д;

- вторинні самописні прилади на щитах керування — на нові моделі заводу «Львівприбор»;

- датчики витрати, рівня, тиску САПФІР — на нові датчики аналогічного призначення САФІР, розроблені харківським ЗАТ «Манометр»;

- загальнопромислові датчики температури — на спеціалізовані датчики Львівського АТ «Термоприбор»; дозволені для застосування на АЕС України.

Більшість компонентів були замінена ще до пуску енергоблоків, заміна інших передбачалася під час наступних планово-запобіжних ремонтів.

Менш відповідальні системи, а також значна частина кабелів, гермопроходки й будівельні конструкції були реалізовані відповідно до проектної АСУТП.

Модернізовані АСУ ТП українських енергоблоків

Для заміни проектної апаратури контролю нейтронного потоку замість АКНП-І застосовані програмно-технічні комплекси АКНП-ІФ (розроблювач і виготовлювач — СНПО «Імпульс») з більш досконалими пристроями детектируванняя нейтронного потоку на основі іонізаційних камер французької фірми Photonis;

У керуючих системах безпеки (технологічних захистів і блокувань) передбачена в проектній АСУ ТП апаратура агрегатних комплексів УКТС і «Каскад-2» замінена програмно-технічними комплексами ПТК УСБ (розроблювач і виготовлювач — НВП «Радій»), які виконують також інформаційні функції моніторингу, відображення, сигналізації, архівування й реєстрації;

На енергоблоках Південно-Української АЕС при модернізації керуючої обчислювальної системи замість ІВС « Комплекс-АЕС» застосований програмно-технічний комплекс ПТК ІВС , реалізований на платформах «Вулкан» і « Вулкан-М» (розроблювач і виготовлювач — ТОВ «Вестрон»);

З ПТК ІВС інтегрована нова система виставлення параметрів безпеки (СВПБ), реалізована на платформі WDPF-II [2.40] (розроблювач і виготовлювач ТОВ «Вестрон» разом з Westinghouse Electric Company);

Системи регулювання рівня живильної води в парогенераторах, реалізовані згідно із проектом на базі агрегатного комплексу технічних засобів «Каскад-2»,

замінені програмно-технічним комплексом СУПВ (розроблювач і виготовлювач Westinghouse Energy Europe, за участю ТОВ «Вестрон» і ВАТ «ЛьвівОРГРЕС»).

Модернізована системауправління й захисту

СУЗ утворена взаємозалежними інформаційними й керуючими системами, що виконують функції контролю нейтронного потоку; аварійного й попереджувального захисту; автоматичного регулювання, розвантаження й обмеження потужності реактора; керування органами регулювання.

Контроль нейтронного потоку

Функції контролю нейтронного потоку в складі СУЗ виконують два незалежні комплекти апаратури АКНП. Основу кожного комплекту утворює програмно-технічний комплекс (ПТК АКНП-ІФ), разом з яким поставляється периферійне устаткування — датчики (пристрою детектирування нейтронного потоку), пристрою визначення граничних значень (уведення уставок), засобу сигналізації й відображення даних. У кожному комплекті можна виділити три, що резервують один одного незалежних канали. Канал включає: три пристрої детектирування нейтронного потоку (УД) відповідно для піддіапазонів, контрольованих при перевантаженні ядерного палива, пуску реактора (пусковий) і роботі на потужності (робітник); центральний пристрій нагромадження й обробки (УНО); пристрій уведення уставок і засобу сигналізації, розташовані в приміщенні блокового щита керування. До складу кожного каналу входить також пристрій (який забезпечує можливість ручного керування механізмами переміщення й індикацію положень блоків детекторування, що контролюють нейтронний потік при перевантаженні ядерного палива. Спільними для трьох каналів одного комплекту є пристрої реєстрації й відображення (ПРВ), сигналізатор оптико-акустичний (СОА), дисплей функціональний (ДФ) і символьний індикатор, що перебувають на БЩУ, а також сигналізатор оптико-акустичний, розміщений на пульті системи керування машини перевантажувальної (СКМП).

УНО кожного каналу: приймає від УД кодовані сигнали, відповідні до значення щільності нейтронного потоку (нейтронної щільності реактора);

обчислює відносну потужність P (у відсотках номінального значення сили

реактора), період зміни сили *Т* і реактивність;

порівнює обчислені значення відносної потужності й періоду із заданими граничними значеннями (уставками).

Передбачається завдання уставок аварійного захисту по потужності (АЗ *«Р»)* і періоду (АЗ « *Т»)* і попереджувального захисту по потужності (ПЗ «*Р*») і періоду (ПЗ *« Т»)* окремо для контрольованого при перевантаженні, пускового й робочого піддіапазонів, а також верхньої межі регулювання потужності (РМ *«Р>),* яка відповідає 102 % відносній потужності реактора.

Якщо відносна потужність або період виходять за задані для них граничні значення (тобто при виконанні будь-якої з умов *Р* > АЗ *«Р», Т* < АЗ «7», *Р* > ПЗ *«Р>, Т* < ПЗ «7>>), на- відповідних виходах каналу формуються дискретні сигнали, що надходять на входи системи, яка реалізує функції аварійного й попереджувального захисту. Безперервні сигнали, що представляють значення відносної потужності *Р,* передаються з виходів кожного каналу першого й другого комплекту АКНП на входи ПТК АРМ-РОМ-УПЗ, який реалізує функції автоматичного регулювання, розвантаження й обмеження потужності реактора. На той же ПТК надходять дискретні сигнали, формовані в канатах АКНП за умови, що відносна потужність перевищує 75 % номінального значення *(Р* > 75 %), а також у випадку, коли відносна потужність реактора досягає верхньої межі регулювання потужності *(Р* > РМ *«Р»),*

В АКНП формуються також вихідні сигнали:

-для системи внутрішньореакторного контролю (СВРК) і інформаційно-обчислювальної системи (ІОС) енергоблоку — безперервні,, що представляють задані (граничні) і поточні (усереднені по трьом каналах) значення відносної потужності й періоду, і дискретні, що вказують контрольований у цей момент піддіапазон нейтронного потоку (відповідний до перевантаження ядерного палива, пуску реактора або роботі на потужності), а також стан перевірки й справність АКНП;

- для СКМП — дискретні, що сигналізують про перевищення значень відносної потужності або періоду, заданих у якості уставок попереджувального й

аварійного захисту для режиму перевантаження ядерного палива: СТОП (при $P >$ ПЗ *«Р»* або $T <$ ПЗ «Г») і РЕВЕРС (при $P >$ АЗ *«Р»* або $T <$ АЗ «Г»),

Поточні значення відносної потужності, періоду й реактивності, обчислені в кожному каналі, а також усереднені по трьом каналам, відображаються в цифровій формі на символьному індикаторі й на екрані ДФ (у вигляді оцифрованих графіків і гістограм). Крім них на екрані ДФ відображаються нижні й верхні межі нейтронної потужності в кожному підіапазоні, індикується поточний піддіапазон, сигналізується досягнення кожної із заданих уставок потужності й періоду.

Пристрій реєстрації й відображення (ПРО) ухвалює дані від трьох каналів, зберігає їх в архіві (на жорсткому диску), забезпечує створення (редагування) архівних кадрів, їх перегляд і копіювання на зовнішній носій. Передбачена можливість видачі даних з ПРО в мережу Ethernet, наприклад для передачі повідомлень у СВРК і ІОС.

Аварійний і попереджувальний захист

Функції аварійного й попереджувального захисту виконує система АЗ-ПЗ, що складається із двох незалежних комплектів. До складу кожного комплекту входять: програмно-технічний комплекс ПТК АЗ-ПЗ; датчики температури, тиску, рівня, перепаду тиску, частоти, потужності, стану технологічного устаткування; елементи ручного керування й сигналізації; сполучні кабелі.

Кожний комплект АЗ-ПЗ має три незалежні канали, що резервують один одного відповідно до логічної умови «два із трьох». Для кожного каналу передбачена окрема шафа формування сигналів (ШФС), що виконує всі основні функції аварійного й попереджувального захисту. Кожний канал має повний набір необхідних датчиків, незалежних від датчиків інших канатів.

Загальними для трьох каналів одного комплекту ПТК АЗ-ПЗ є: кросова вихідна шафа, елементи (ключі) ручного керування й табло сигналізації, установлювані на БЩУ й РЩУ; робоча станція архівування, відображення й реєстрації даних (РС) і автоматизоване робоче місце технолога (АРМТ), розташовані в приміщенні чергового персоналу ЦТАІ.

Кожний канал (шафа ШФС) першого й другого комплектів ПТК АЗ-ПЗ

ухвалює: безперервні й дискретні сигнали від датчиків цього каналу; дискретні сигнали *P* > АЗ *«P», T*< АЗ « *T», P* > ПЗ *«P», T*< ПЗ *«T»* від відповідного комплекту й каналу АКНП; сигнали від системи внутрішньореакторного контролю (мінімальний, що допускається запас до кризи кипіння, перевищення локального енерговиділення, перевищення температури теплоносія в локальній зоні); сигнали від системи електроживлення (відсутність живлячих напруг на фідерах); команди від ключів керування на БЩУ й РЩУ. Крім того, на вхід кожного каналу надходить дискретний сигнал від іншого комплекту ПТК АЗ-ПЗ при виводі цього комплекту з роботи, наприклад для перевірки або технічного обслуговування.

Канал, який або виявив відхилення хоча б одного контрольованого (вимірюваного або розрахункового) параметра за кордон уставки спрацьовування аварійного захисту, або ідентифікував порушення якого-небудь із заданих умов безпечної експлуатації, або одержав сигнал *P* > АЗ«Р», *T*< АЗ« Г» від АКНП або команду оператора (від ключа «АЗ» на БЩУ або РЩУ), формує дискретний сигнал, який передається в кросову вихідну шафу. На підставі цих сигналів, отриманих, принаймні, від двох ШФС, кросова вихідна шафа формує відповідно до прийнятої логічної умови й видає команду аварійного захисту (АЗ):

-на входи трьох канатів виконавчої системи (СГИУ), яка безпосередньо управляє органами регулювання ОР (ініціює аварійну зупинку реактора за рахунок зняття із усіх ОР основного й резервного живлення);

-на перший і другий входи передбаченої проектом системи силового електроживлення приводів органів регулювання ССП (ініціює аварійну зупинку реактора за рахунок відключення напруги змінного струму на обох уведеннях силового електроживлення, від яких одержують енергію всі ОР);

-на входи трьох канатів автоматичної системи керування турбіною АСУТ ініціює дії, що приводять до розвантаження турбіни при зупинці реактора);

-на вхід системи борного регулювання СБР (ініціює включення насоса подачі бору високого тиску).

Одночасно на одному з виходів кросової шафи формується сигнал, що викликає першопричину спрацьовування АЗ (включає відповідне табло на БЩУ).

Команда АЗ і сигнал первопричини спрацьовування зберігаються на виході кросової шафи доти, поки вони не будуть скинуті оператором за допомогою відповідних ключів на БЩУ.

Подібним же чином кросова вихідна шафа формує й видає команду попереджувального захисту (ПЗ-1 або ПЗ-2) на підставі сигналів, отриманих принаймні від двох або від усіх трьох каналів одного комплекту, що виявили відхилення якого-небудь контрольованого параметра за кордон уставки спрацьовування ПЗ-1 або ПЗ-2, або ідентифікувавших порушення умови нормальної експлуатації, або, що одержали сигнал *P* > ПЗ *«P», T*< ПЗ «7» від АКНП або команду оператора (від ключа «ПЗ» на БЩУ).

Команда ПЗ-1 надходить: на входи трьох каналів виконавчої системи, що безпосередньо управляє органами регулювання (ініціює зменшення потужності реактора за рахунок послідовного опускання груп ОР в активну зону із заданою швидкістю);

на входи трьох каналів АСУТ (ініціює дії, що приводять до зменшення потужності турбіни);

на входи трьох каналів системи, яка реалізує функції автоматичного регулювання, розвантаження й обмеження потужності реактора (забороняє роботу автоматичного регулятора, який міг би протидіяти зменшенню потужності).

Команда ПЗ-2, що надходить від кросової шафи на аналогічні входи тих же систем, забороняє будь-які дії цих систем, які могли б викликати збільшення потужності реактора або турбіни.

Одночасно з видачею команди ПЗ- 1 або ПЗ-2 на виході кросової шафи формується сигнал першопричини спрацьовування, який включає відповідне табло на БЩУ. Команди ПЗ-1 і ПЗ-2 видаються доти, поки зберігаються их порушення, що викликали; сигнал першопричини спрацьовування знімається ключем КВІТИРУВАННЯ.

Аналогічні сигнали й команди формуються другим комплектом АЗ-ПЗ, при цьому всі зазначені вище дії сполучених систем можуть бути ініційовані командами будь-якого комплекту. Поточна, діагностична й архівна інформація від

обох комплектів передається в інформаційно-обчислювальну систему (ІОС) енергоблоку.

Автоматичне регулювання, розвантаження, обмеження потужності

Функції автоматичного регулювання, розвантаження й обмеження потужності реактора, а також прискореного попереджувального захисту (прискореного розвантаження блоку) виконує система АРМ-РОМ-УПЗ, до складу якої входять датчики теплотехнічних параметрів, подій, станів і програмно-технічний комплекс ПТК АРМ-РОМ-УПЗ із робочою станцією (РС) і панелями сигналізації й керування, розташованими на БЩУ.

Система має три незалежні канали, що резервують один одного відповідно до логічної умови «два або більш із трьох». Кожний канал має повний набір необхідних датчиків, незалежних від датчиків інших каналів. Для кожного каналу передбачена окрема шафа формування сигналів (ШФС), що складається із двох частин. Одна з них реалізує функції автоматичного регулювання потужності, інша керує розвантаженням, обмеженням потужності реактора й прискореним попереджувальним захистом. Кросова вихідна шафа , робоча станція й панелі сигналізації й керування являються спільними для трьох каналів.

Розташована в ШФС частина каналу, яка реалізує функції автоматичного регулювання потужності (АРМ), подає:

безперервні сигнали від датчиків цього каналу, що представляють тиск над активною зоною й у головному паровому колекторі;

безперервні сигнали, що представляють значення відносної потужності *P,* і дискретні сигнали *P* > РМ«Р» від першого й другого комплекту АКНП;

команди ПЗ-1 (заборона регулювання) і ПЗ-2 (заборона збільшення потужності) від першого й другого комплекту АЗ-ПЗ, а також дискретний сигнал про вивід з роботи відповідного комплекту;

дискретні сигнали від панелі сигналізації й керування, установленої на БЩУ, що визначають обраний режим керування;

дискретні сигнали від перемикача, установленого на БЩУ, що визначають обраний режим роботи (автоматичний або дистанційний).

Дискретні вихідні сигнали АРМ, формовані при відхиленні регульованого параметра (відносної потужності або тиски в головному паровому колекторі) від усталовленого значення, ініціюють дії виконавчої системи, що управляє органами регулювання, так, щоб мінімізувати це відхилення за рахунок підйому або опускання в активну зону робочої групи органів регулювання. У систему регулювання турбіни передаються із кросової шафи сигнали, що визначають обраний режим роботи й режим керування АРМ. Аналогічна інформація з кожного каналу, а також дані про видавані команди регулювання, викликувані ними дії, забороні зменшення й (або) збільшення потужності, стані (несправності) кожного каната індикуються на панелі сигналізації й керування, установленої на БЩУ). Загальні сигнали про несправність і вивід з роботи будь-якого каналу управляють включенням табло на БЩУ.

Розташована в ШФС частина каналу, яка реалізує функції розвантаження й обмеження потужності (РОМ) і прискореного попереджувального захисту | УПЗ), приймає:

безперервні сигнали від датчиків, що й нормують перетворювачів цього каналу, що представляють температуру теплоносія, тиск, частоту живлення й потужність ГЦН;

дискретні сигнали від датчиків цього каналу, що представляють стан турбогенератора (відключення, посадка стопорних клапанів, зняття навантаження генератора);

безперервні сигнали, що представляють значення відносної потужності, і дискретні сигнали $P > 75$ % від першого й другого комплекту АКНП;

сигнали про вивід з роботи першого й другого комплекту АЗ-ПЗ;

команди від ключів, розташованих на БЩУ.

На підставі сигналів, отриманих від двох чи трьох каналів, у яких виявлено виникнення заданих умов спрацьовування РОМ, кросова вихідна шафа формує й видає команду розвантаження на входи трьох каналів виконавчої системи, що управляє органами регулювання (ініціює зменшення потужності реактора за рахунок послідовного опускання ОР в активну зону). Дані про спрацьовування,

перевірку, готовність і несправності кожного каналу индикуються на панелі сигналізації, установленої на БЩУ. Загальні сигнали про несправність, вивід з роботи будь-якого каналу, спрацьовуванні РОМ, стані устаткування, що викликав розвантаження, доступі усередину будь-якої шафи управляють включенням відповідних табло на БЩУ.

При відключенні основного устаткування, посадці стопорних клапанів турбіни, відключенні блоку від енергосистеми на потужності, що перевищує 75 % номінального значення, ішли при безпосередньому введенні команди УПЗ ключем, розташованим на БЩУ, кросова вихідна шафа формує (на підставі сигналів, отриманих від двох або трьох каналів) і видає команду прискореного попереджувального захисту (УПЗ):

на входи трьох каналів виконавчої системи, що безпосередньо управляє органами регулювання (ініціює швидке зменшення потужності реактора за рахунок зняття основного й резервного живлення із усіх ОР заздалегідь певної групи і її падіння в активну зону);

на входи АСУТ (ініціює дії, що викликають відповідне зменшення потужності турбіни).

Одночасно на виходах кросової шафи формуються сигнали для включення табло на БЩУ, індикуючі спрацьовування УПЗ і першопричину спрацьовування. Команда УПЗ і сигнал першопричини зберігаються на виході кросової шафи доти, поки вони не будуть скинуті оператором за допомогою відповідних ключів на БЩУ. Поточна, діагностична й архівна інформація формується робочою станцією й передається в ІВР енергоблоку.

Керування органами регулювання

Функції безпосереднього керування органами регулювання виконує система групового й індивідуального керування (СГІК). Керування органами регулювання здійснюється у всіх режимах роботи енергоблоку на потужності, а також при планових і аварійних зупинках реактора й передбачає вплив на протікання ланцюгової реакції в активній зоні з метою підтримки параметрів реакторної установки в заданих межах, зміни параметрів по заданих алгоритмах, припинення

ланцюгової реакції, підтримки підкритичного стану реактора протягом як завгодно тривалого часу.

При роботі енергоблоку на потужності здійснюється зміна положення в активній зоні декількох органів регулювання, що утворюють групу (групове керування), або одного органу регулювання (індивідуальне керування). При плановій зупинці, яка проводиться з метою ремонту устаткування або перевантаження ядерного палива, усі органи регулювання вводяться в активну зону, що викликає припинення ланцюгової реакції. Аварійна зупинка ініціюється командою аварійного захисту, отриманої від будь-якого комплекту системи АЗ-ПЗ. При цьому із усіх приводів знімається силове живлення й органи регулювання падають в активну зону під дією власної ваги, що приводить до швидкого припинення ланцюгової реакції й зупинки реактора.

СГІК забезпечує автоматичне й ручне (дистанційне) керування органами регулювання. Автоматичне керування здійснюється: по командах аварійного захисту (АЗ), попереджувального захисту (ПЗ-1), прискореного попереджувального захисту (ППЗ), розвантаження реактора (РОМ) і автоматичного регулювання потужності (АРМ).

Ручне (дистанційне) керування виконується з ініціативи оператора:

по команді підйому або опускання з робочою швидкістю кожної (однієї) групи органів регулювання або переміщення груп одна за іншою в проектній послідовності;

по команді підйому або опускання з робочою швидкістю кожного (одного) органу регулювання;

по команді підйому або опускання з робочою швидкістю однієї (п'ятої) групи органів регулювання.

До складу системи входять: датчики положення органів регулювання; виконавчі механізми (крокові електромагнітні приводи КЕП); програмно-технічний комплекс ПТК СГІК з робочою станцією РС, установленої на робочому місці чергового інженера ЦТАІ, панеллю контролю й керування (ПКК), розташованою в приміщенні БЩУ, і пристроями індикації положень ОР, які

розміщаються на БЩУ й РЩУ.

Функції формування команд аварійного й прискореного попереджувального захисту, а також команд групового й індивідуального керування ОР виконують три ідентичні незалежні канали (шафи 1ШФС...ЗШФС), що резервують один одного. Кожний канал виконує:

команди аварійного (АЗ) і попереджувального захисту (ПЗ-1, ПЗ-2) від відповідних каналів першого й другого до: комплекта ПТК АЗ-ПЗ;

команди розвантаження реактора (РОМ), • е корінного попереджувального захисту >713) і регулювання потужності (АРМ) : г відповідних канатів ПТК АРМ-РОМ-УПЗ;

номер групи ОР, обраної для ручного керування, і команди підйому чи спуску обраної групи від перемикача ВИБІР ГРУПИ й ключа ГРУПОВЕ КЕРУВАННЯ (або від ключа КЕРУВАННЯ 5 ГРУПОЮ) на ПКУ:

координати ОР, обраного для ручного керування, і команди підйому або спускання обраного ОР від кнопок складального поля й ключа ІНДИВІДУАЛЬНЕ КЕРУВАННЯ на ПКУ);

дані про положення всіх ОР по висоті активної зони, знеструмленні електромагнітів (падінні ОР) і тривалості падіння у вигляді цифрових повідомлень від шаф контролю положення 1ШКП...4ШКП.

Від кожного з каналів (шаф 1ШФС...ЗШФС) передаються:

у шафи силового керування приводами 1ШСУ...16ШСУ — керуючі сигнали, що ініціюють зняття силового живлення із усіх приводів (по команді АЗ) або із заздалегідь обраної групи приводів (по команді УПЗ);

у шафи 1ШСУ...16ШСУ — сигнали, що управляють переміщенням групи ОР або окремого ОР у ручному режимі (по командах від ПКУ) або в автоматичному режимі (по командах ПЗ-1, РОМ, УПЗ, АРМ);

у ПКУ — дані про положення по висоті всіх ОР і неузгодженості ОР у кожній групі, а також тривожне повідомлення у випадку перевищення, що допускається неузгодженості хоча б в одній групі.

Канали силового керування в шафах 1ШСУ...16ШСУ здійснюють

безпосереднє керування приводами ОР. Кожний канал:

приймає сигнали від шаф 1ШФС... ЗШФС, ініційовані командами АЗ і УПЗ, і знеструмлює електромагніти керованого привода при одержанні сигналів, принаймні, від двох шаф ШФС;

приймає повідомлення від кожного із шаф ШФС, здійснює обробку прийнятих даних відповідно до логічної умови «два із трьох», формує та видає на привод ШЕМ послідовність імпульсів, що викликають переміщення ОР (або втримує ОР нерухомим при відсутності керівних сигналів, принаймні, від двох ШФС, а також при досягненні органом регулювання крайніх положень);

автоматично перемикає привід на живлення від резервного джерела при зникненні основного електроживлення або несправностях у ланцюгах силового керування.

Під час дії команди ПЗ-2 від першого або другого комплекту ПТК АЗ-ПЗ видача будь-яких сигналів на рух ОР нагору блокується.

На панелі індикації й моніторі панельного комп'ютера, вбудованих у ПКУ, відображаються номер керованої групи, координати ОР, обраного для індивідуального керування, направлення переміщення, положення по висоті й інші дані по викликові оператора. Вбудовані засоби діагностики здійснюють:

безперервний автоматичний контроль технічного стану всіх складових частин ПТК СГІК, сполученого периферійного встаткування й ліній передачі сигналів і повідомлень;

обробку отриманих даних, архівування, відображення (безупинно й по викликові оператора) поточної й архівної інформації, видачу діагностичних повідомлень в ІВС і СВРК;

звукову сигналізацію при виявленні відмов і несправностей, вивід відповідних тривожних

Висновки

Кожна ІКС призначена для конкретного енергоблоку й повинна була враховувати вимоги замовника (АЕС) технології, що випливають із особливостей

технологічних систем і устаткування, зв'язків з іншими ІКС за станом на момент впровадження, особливостей роботи оперативного персоналу й т.п.

Сучасні комп’ютерні системи розроблені згідно нових вимог до ядерної та радіаційної безпеки з метою модернізації та підвищення безпеки роботи енергоблоків водо-водяних енергетичних реакторів

2.3. Оцінка ефективності процесів життєвого циклу бізнес критичних програмних систем управління економічним обєктом

Для оцінки ефективності процесів життєвого циклу бізнес критичних програмних систем управління економічним обєктом можна звернутися до даних IoT Analytics серед яких в 2016 році найбільше проектів (22% від загальної кількості), пов'язаних із застосуванням інтернету речей, було реалізовано для промислових об'єктів. Це підтверджує розвиток і поширення технологій заявлених в доктрині Industry 4.0.

Таким чином, на наших очах виник новий клас кібер-фізичних систем, що отримав назву Industrial Internet Control Systems (IICS) або Industrial Internet of Things (IIoT).

З назви зрозуміло, що такі системи є гібридом технологій, застосовуваних в АСУ ТП і в системах на базі інтернету речей. Відповідно в таких системах необхідно враховувати всі ризики, пов'язані з порушенням властивостей інформаційної (security) і функціональної безпеки (safety).

При розгляді вимог до інформаційної безпеки АСУ ТП у нас вийшло гармонізувати вимоги до інформаційної безпеки (ІБ) і функціональної безпеки (ФБ). Як видно на малюнку, процеси забезпечення ІБ і ФБ повинні реалізовуватися в рамках життєвого циклу.

Оскільки мова йде, в першу чергу, про АСУ ТП, їх життєвий цикл зазвичай розглядається в прив'язці до промислового об'єкту управління. Тому і розглядається так званий «великий» життєвий цикл, пов'язаний з проектуванням,

будівництвом, експлуатацією та утилізацією промислового об'єкта, де АСУ ТП є однією зі складових. Звичайно, може здійснюватися і модернізація існуючого обладнання АСУ ТП, але в «великий» життєвий цикл(див. Рис. 2.5.) це також вписується.

З цією метою розглянемо вимоги стандарту МЕК 61508 «Функціональна безпека систем електричних, електронних, програмованих електронних, пов'язаних з безпекою» (IEC 61508 Functional safety of electrical / electronic / programmable electronic safety-related systems).

Розглянемо у стандарті МЕК 61508 V-подібний («малий») життєвий цикл. Згідно з цим концепція, специфікація вимог і реалізація описуються так званим V-образним або «малим» життєвим циклом. Іноді концепцію виносять за рамки «малого» життєвого циклу, але ми її включимо, оскільки в ній повинні міститися вихідні положення для формування специфікації вимог.

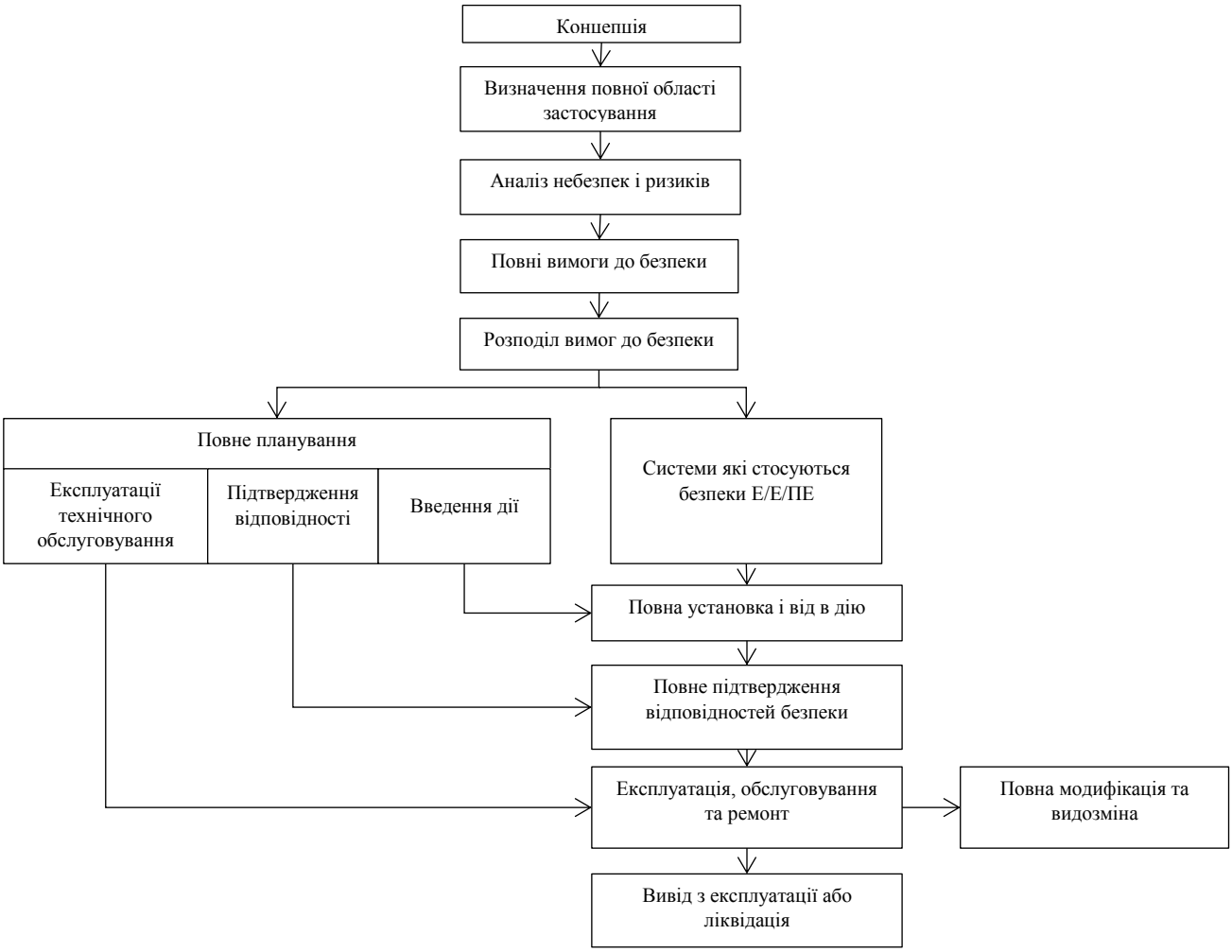


Рис. 2.5. Повний життєвий цикл розробки систем безпеки

Також стандарт МЕК 61508 розкриває структуру етапу «10 E / E / PE system realization». Структура процесу реалізації пропонується як для програмного так і для апаратного забезпечення, причому для останнього різниться програмована складова (наприклад, ПЛІС, програмовані інтегральні схеми) і непрограмовані складові.

Стандарт МЭК 61508 містить кілька неточностей та пробілів, повязаних із структурою життєвого циклу системи. Це повязано з тим, що зроблена спроба в «малому» життєвому циклі розділити вимоги до програмного та апаратного забезпечення, а системний рівень представити у «великому» життєвому циклі.

Таким чином, одна з проблем стандарту МЕК 61508 заключається в тому, що в ньому відсутній в явному вигляді життєвий цикл, який міг би безпосередньо бути

застосованим в комп'ютерних системах керування, будь то АСУ ТП, вбудовані системи чи рівень пристроїв «інтернета речей».

Зазвичай в практиці програмної і системної інженерії йдеться про розробку системи до моменту передачі її постачальнику. Ми будемо детально розглядати саме такий V-подібний життєвий цикл. Структура життєвого циклу визначена стандартами по функціональної безпеки, в тому числі і МЕК 61508 (див. Рис. 2.6.). По низхідній гілці життєвого циклу виконується розробка «зверху - вниз». По висхідній гілці життєвого циклу виконується інтеграція «знизу - вгору», супроводжувана тестуванням на відповідність тим або іншим проектним документам.

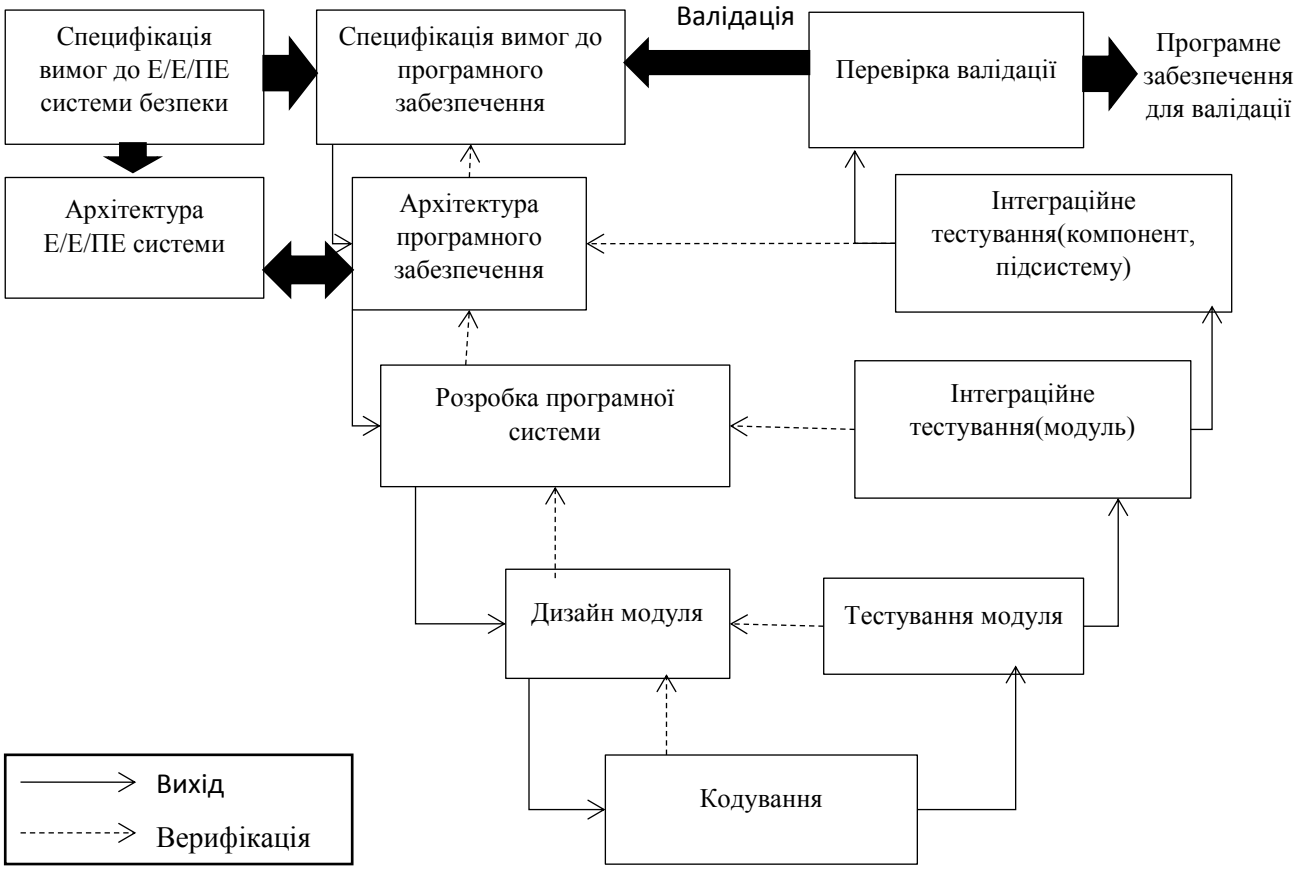


Рис. 2.6. МЕК 61508-3- Систематичні можливості програмного забезпечення та життєвий цикл розробки (V-модель)

Пропонована структура життєвого циклу, з одного боку, не суперечить вимогам МЕК 61508, а з іншого боку закриває деякі прогалини у вимогах,

показуючи, як можна в рамках єдиної моделі інтегрувати розробку, перевірки та затвердження системи, програмного забезпечення та апаратних засобів.

Життєвий цикл інформаційної та функціональної безпеки розробки програмного забезпечення представлено на рис. 2.7.

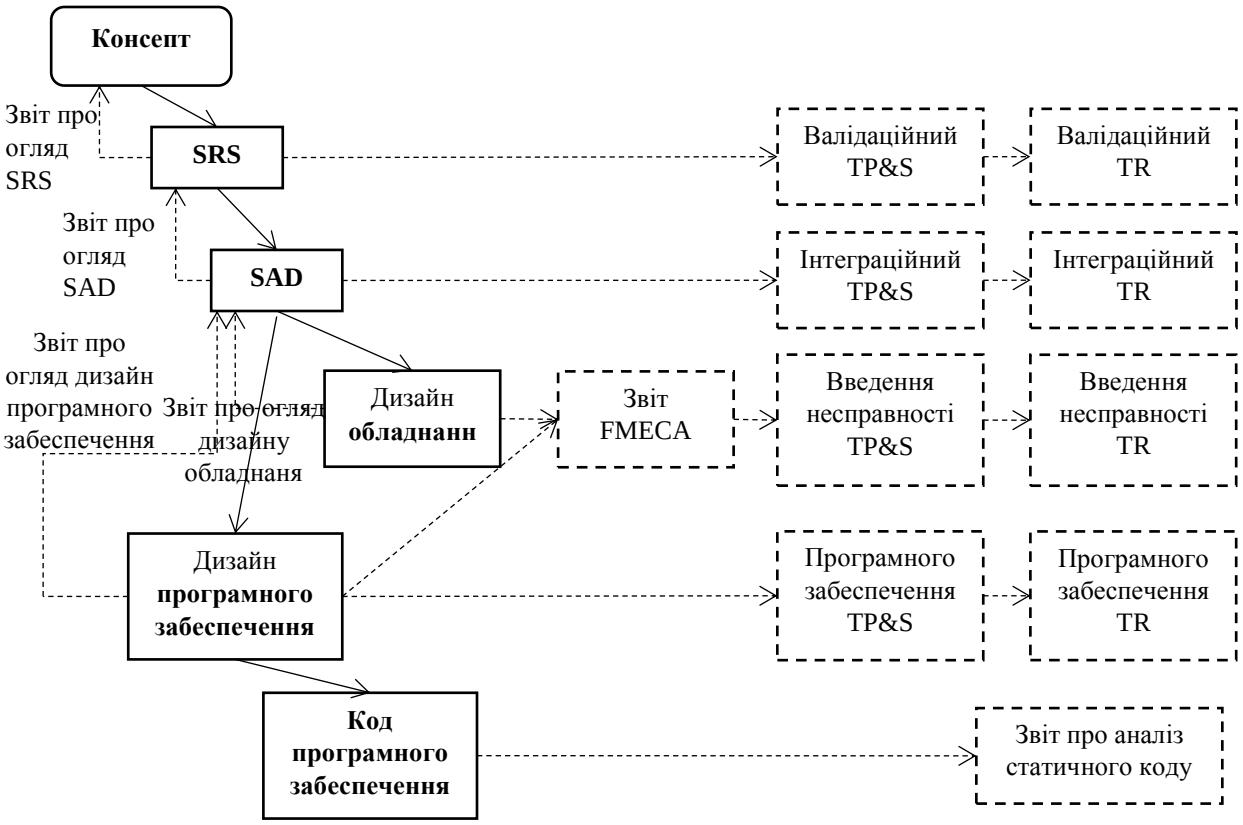


Рис. 2.7. Життєвий цикл інформаційної та функціональної безпеки розробки програмного забезпечення

Модель життєвого циклу включає в себе наступні послідовно виконуються етапи (на Рис. 2.7 етапи позначені іменами підсумкових документів):

-розробка концепції (Concept);

- розробка специфікації вимог з безпеки (Safety Requirements Specification, SRS), яка описує систему у вигляді «чорного ящика», тобто, «що виконується», а не «як виконується»;

- огляд специфікації вимог по безпеки (SRS Review), є етапом процесу верифікації та валідації;

- розробка проекту архітектури системи (System Architecture Design, SAD), яка описує систему у вигляді «білого ящика», тобто, «як виконується», а не «що виконується»;

- огляд проекту архітектури системи (SAD Review), є етапом процесу верифікації і та валідації;

- розробка проекту апаратних засобів (Hardware Design), в українськомовній термінології ця складова називається конструкторської документації (КД), в неї входять, як проекти електронних плат, так і креслення механічних конструкцій і електричної частини (так званої «обв'язки»), що включає кабелі, компоненти електропостачання та сполучення з польовим обладнанням (датчиками і виконавчими механізмами;

- огляд проекту апаратних засобів (Hardware Design Review), є етапом процесу верифікації та валідації;

- аналіз видів, наслідків та критичності відмов (Failure Mode, Effect and Criticality Analysis, FMECA), є етапом процесу верифікації та валідації; при виконанні FMECA в першу чергу враховується структура апаратних засобів, однак, прийматися до уваги також механізми діагностування та відмовостійкості, що реалізуються в програмному забезпеченні;

- проектування програмного забезпечення (Software Design);

- огляд проекту програмного забезпечення (Software Design Review), є етапом процесу верифікації та валідації;

- розробка коду програмного забезпечення (Software Coding);

- статичний аналіз програмного коду (Static Code Analysis);

- тестування програмного забезпечення на відповідність вимогам проекту (Software Testing) включає в себе як юніт-, так і інтеграційне тестування, як функціональне, так і структурне тестування; перед початком тестування розробляється план і специфікація тестування програмного забезпечення (Software Test Plan and Specification, TP & S), результати тестування документуються в звіті про тестування програмного забезпечення (Software Test Report, TR);

- тестування методом засіву дефектів в апаратні засоби і програмне забезпечення (Fault Insertion Testing), входом для якого є результати FMEDA в частині аналізу реалізації самодіагностики; перед початком тестування розробляється план і специфікація тестування методом засіву дефектів (Fault Insertion TP & S), результати тестування документуються в звіті про тестуванні методом засіву дефектів (Fault Insertion TR);

- інтеграційне тестування інтегрованих програмно-апаратних компонентів на відповідність вимогам до архітектури (Integration Testing) ; перед початком тестування розробляється план і специфікація інтеграційного тестування (Integration TP & S), результати тестування документуються в звіті про інтеграційний тестуванні (Integration TR);

- валідаційні тестування інтегрованої системи на відповідність специфікації вимог з безпеки (Validation Testing); перед початком тестування розробляється план і специфікація валідаційні тестування (Validation TP & S), результати тестування документуються в звіті про валідаційні тестуванні (Validation TR); враховуючи важливість даного заключного етапу життєвого циклу, МЕК 61508 вимагає, щоб план валідаційні тестування було складено відразу після закінчення розробки специфікації вимог з безпеки (SRS); валідація може включати в себе крім функціонального тестування також тестування на стійкість до екстремальних впливів навколишнього середовища (температурно-вологісним, механічним, електромагнітним, радіаційним і т.д.).

Детальний опис фаз життєвого циклу представлено у вигляді набору таблиць. У таблицях синхронізована діяльність із забезпечення як функціональної, так і інформаційної безпеки. Деякі дії можуть бути віднесені одночасно і до ФБ, і ІБ.

Для комп'ютерних систем управління основні ризики відносяться до порушень властивості ФБ. Тому в таблиці пріоритет відданий ФБ. Ті дії, які забезпечують і ФБ, і ІБ внесені в графу, що стосується ФБ. Відповідно, в графі, що відноситься до ІБ враховані дії, які не перекриваються діяльністю щодо забезпечення ФБ.

Таблиця 2.1.

Фаза Concept

Найменування фази	Виконувані завдання	Забезпечення функціональної безпеки	Забезпечення інформаційної безпеки
Концепція (Concept)	Розробка концептуального документа верхнього рівня, у якому визначаються потреби підприємств або бізнесу й автоматизація процесів, включаючи ідентифікацію небезпек і погроз	Визначити місію, бачення й цінності забезпечення ФБ Ідентифікувати потреби в забезпеченні ФБ для цільових або ринкових галузей і клієнтів, а також описати продукт, який буде задовольняти ці потреби, і що розроблювач готовий надати Документувати внутрішні й зовнішні небезпеки для підприємства Установити рівні повноти безпеки (Safety Integrity Levels, SIL) для систем і компонентів Розробити політики ФБ для комп'ютерних систем керування й устаткування, мереж, інформаційних систем і персоналу Визначити нормативні вимоги й стандарти, застосовувані для комп'ютерних систем керування Документувати активи, сервіси й персонал, що бідують у рівнях захисти	Визначити місію, бачення й цінності забезпечення ІБ Визначити потреби в захисті майна, активів, сервісів або персоналу Документувати потенційні внутрішні й зовнішні погрози для підприємства Установити рівні захищеності (Security Levels, SL) для систем і компонентів Розробити політики ІБ для комп'ютерних систем керування й устаткування, мереж, інформаційних систем і персоналу

Таблиця 2.2.

Фаза Safety Requirements Specification

Найменування фази	Виконувані завдання	Забезпечення функціональної безпеки	Забезпечення інформаційної безпеки
Специфікація вимог по безпеці (Safety Requirements Specification,SRS)	Розробка системної специфікації функціональних вимог і вимог по безпеці (методологія «чорний ящик», коли описується, «що виконується», а не « як виконується), включаючи режими, тимчасові характеристики, інтерфейси, сигнали, самодіагностику, періодичне тестування, граничні зовнішні умови й ДР-	Розробити план керування функціональною безпекою (Functional Safety Management Plan) *Примітка. Планування може бути організоване як окремий етап до розробки специфікації* Виконати оцінку збитку об'єктів і пов'язаних з ними сервісів для переліку потенційних небезпек Провести аналіз ризиків, потенційного збитку й небезпек Класифікувати ризики, потенційні наслідки для підприємства й бізнесу, а також методи зниження ризиків Установити вимоги до ФБ (у тому числі вимог до функцій безпеки й вимог до повноти безпеки) для комп'ютерних систем керування й устаткування, мереж, інформаційних систем, а також персоналу Розподілити функції між керованими завданнями й модулями для розробки дизайну архітектури Підготувати документ для трасування вимог, розробивши прозору структуру, засновану на вимогах з розміщенням тегів	Розробити програму інформаційної безпеки (Security Program) Виконати оцінку оцінки уразливостей об'єктів і пов'язаних з ними сервісів для переліку потенційних погроз Провести аналіз ризиків потенційних уразливостей і погроз Установити вимоги до ІБ для комп'ютерних систем керування й устаткування, мереж, інформаційних систем і персоналу

Таблиця 2.3.

Фаза Safety Requirements Specification Review

Найменування фази	Виконувані завдання	Забезпечення функціональної безпеки	Забезпечення інформаційної безпеки
Огляд специфікації вимог по безпеці (Safety Requirements Specification Review)	Верифікація вимог Специфікації (SRS) на відповідність вимогам Концепції (Concept)	Перевірити зрозумілість, точність, недвозначність, верифікаційність, тестованість, супровід і реалізація вимог Перевірити відповідність Специфікації (SRS) вимогам IEC 61508 Перевірити відповідність Специфікації (SRS) вимогам Концепції (Concept) (формальне трасування вимог не є обов'язковим) Перевірити реалізацію функціональних вимог і вимог до повноти безпеки Перевірити наявність Плану валідаційного тестування (Validation Test Plan), розробленого на основі трасування вимог Специфікації (SRS)с	Перевірити відповідність специфікації (SRS) вимогам IEC 62433

Таблиця 2.4.

Фази System Architecture Design, System Architecture Design Review

Найменування фази	Виконувані завдання	Забезпечення функціональної безпеки	Забезпечення інформаційної безпеки
Проект системної архітектури (System Architecture Design, SAD)	Розробки специфікації вимог до системної архітектури (методологія «білий ящик», коли описується, « як виконується», а не «що виконується), включаючи детальну структуру й опис поведінки	Визначити внутрішню структуру системи, забезпечуючи проектні розв'язки для відповідності вимогам, і розподіляючи для цього різні апаратні й програмні компоненти (модулі) системи Указати, які вимоги будуть виконуватися програмним забезпеченням (буде реалізований в Software Design) і які вимоги будуть виконані за допомогою апаратних засобів (будуть реалізовані в Hardware Design) Розподілити вимоги до функцій безпеки й вимоги до повноти безпеки між частинами системи Підготувати документ для трасування вимог, розробивши прозору структуру, засновану на вимогах з розміщенням тегів Виконати організаційні завдання, такі, як розробка програм навчання персоналу й програм керування активами	Визначити функціональні вимоги ІБ для зон підприємства, виробничих потужностей, зон контролю, а також границі й портали контролю доступу Розробка фізичних і логічних систем для виконання функціональних вимог, певних раніше з метою ІБ Визначити організаційну складову для забезпечення ІБ, випустити політики й процедури ІБ
Огляд проекту системної архітектури (System Architecture Design Review)	Верифікація вимог Проекту (SAD) на відповідність вимогам Специфікації (SRS)	Перевірити зрозумілість, точність, недвозначність, верифікаційність, тестованість, супровід і реалізація вимог Перевірити відповідність Проекту (SAD) вимогам IEC 61508 Перевірити відповідність Проекту (SAD) вимогам Специфікації (SRS) (Concept) (потрібна пряма й зворотне трасування вимог)	Перевірити відповідність вимогам IEC 62433

Таблиця 2.5.

Фази Hardware Design, Hardware Design Review

Найменування фази	Виконувані завдання	Забезпечення функціональної безпеки	Забезпечення інформаційної безпеки
Проект апаратних засобів (Hardware Design)	Розробка документації для апаратних засобів (конструкторська документація), на підставі якої здійснюється виробництво	Розробити схеми друкованих плат, специфікації матеріалів і компонентів, механічні й електричні креслення, креслення джерел живлення й комунікаційних пристроїв і т.п. *Примітка. Виробництво апаратних засобів, як правило, не розглядається в V-Образному життєвому циклі, оскільки ставиться до системи менеджменту якості. Однак, виробництво також може бути розглянуте, як етап життєвого циклу*	
Огляд проекту апаратних засобів (Hardware Design Review)	Верифікація Проекту апаратних засобів (Hardware Design) на відповідність вимогам Проекту (SAD)	Перевірити зрозумілість, точність, недвозначність, верифікаційність, тестованість, супровід і реалізація Hardware Design Підготувати документ для трасування вимог, розробивши прозору структуру, засновану на вимогах з розміщенням тегів *Примітка.* Оскільки ставити теги трасування в кресленнях не представляється можливим, трасування вимог виконується для Hardware Design Review Report Перевірити відповідність Hardware Design відповідає вимогам Проекту (SAD) (потрібна пряма й зворотне трасування вимог)	Перевірити відсутність недокументованість можливостей і потенційно шкідливих частин в апаратних засобах

Таблиця 2.6.

Фаза Failure Mode, Effect and Criticality Analysis

Найменування фази	Виконувані завдання	Забезпечення функціональної безпеки	Забезпечення інформаційної безпеки
Аналіз видів, наслідків і критичності відмов (Failure Mode, Effect and Criticality Analysis, FMECA)	Виконання FMECA для Проекту апаратних засобів (Hardware Design) і Проекту програмного забезпечення (Software Design)	Підготувати перелік апаратних компонентів згідно зі специфікацією Визначити інтенсивність відмов для апаратних компонентів Перевірити істинність допущення Про постійну інтенсивність відмов апаратних компонентів у часі (експонентний закон розподілу часу до відмови) Класифікувати можливі відмови апаратних засобів, як діагностованими безпечні, діагностованими небезпечні, недіагностованими безпечні, недіагностованими небезпечні Визначити функції самодіагностики по виявленню апаратних і програмних відмов Розрахувати показники безпеки, необхідні для відповідності цільовому рівню повноти безпеки (Safety Integrity Levels, SIL) Зробити висновок про відповідність або невідповідність заданому рівню SIL. У випадку, коли заданий рівень SIL не досягнуться, повинні бути сформульовані рекомендації з виконання вимог до заданого рівня SIL	Визначити функції самодіагностики по виявленню атак

Таблиця 2.7.

Фази Software Design, Software Design Review, Software Coding, Static Code Analysis

Найменування фази	Виконувані завдання	Забезпечення функціональної безпеки	Забезпечення інформаційної безпеки
Проект програмного забезпечення (Software Design)	Розробка документації для програмного забезпечення, на підставі якої здійснюється кодування	Розробка алгоритмів і структур даних для кожного програмного модуля *Примітка. Звичайно ця стадія включає, по-перше, проект архітектури програмного забезпечення й, в- других, детальний проект програмних модулів і бібліотек*	
Огляд проекту програмного забезпечення (Software Design Review)	Верифікація Проекту програмного забезпечення (Software Design) на відповідність вимогам Проекту (SAD)	Перевірити зрозумілість, точність, недвозначність, верифікаційність, тестованість, супровід і реалізація Software Design Перевірити відповідність Software Design відповідає вимогам Проекту (SAD) (потрібна пряма й зворотне трасування вимог)	
Кодування програмного забезпечення (Software Coding)	Написання вихідного програмного коду	Розробка програмного коду відповідно до вимог Software Design і Посібника з кодування програмного забезпечення (формальне трасування вимог не є обов'язковим) *Примітка. Посібник з кодування програмного забезпечення, як правило, є документом системи менеджменту якістю*	
Статичний аналіз коду (Static Code Analysis)	Верифікація програмного коду на відповідність вимогам до Software Design, включаючи статичний аналіз коду статичний аналіз коду	Виконати перевірку програмного коду й / або статичний аналіз коду без запуску програмного забезпечення на відповідність критеріям Посібника з кодування програмного забезпечення Усунути або дозволити аномалії *Примітка. Для статичного аналізу коду повинні використовуватися комп'ютерні інструменти*	Перевірити відсутність не документованих можливостей і потенційних шкідливих компонентів у програмному забезпеченні

Таблиця 2.8.

Фаза Software Testing

Найменування фази	Виконувані завдання	Забезпечення функціональної безпеки	Забезпечення інформаційної безпеки
Тестування програмного забезпечення (Software Testing)	Інтеграція програмного забезпечення Функціональне й структурне тестування програмного коду на відповідність вимогам Software Design	Підготувати План тестування програмного забезпечення (Software Test Plan) на основі трасування вимог Software Design *Примітка. Юніт і інтеграційне тестування можна розглядати як дві послідовні частини тестування й інтеграції програмного забезпечення* Підготувати Специфікацію тестування програмного забезпечення (Software Test Specification) шляхом розробки тест кейсів для кожної тестованої вимоги Виконати функціональне тестування програмного забезпечення Перевірити тестове покриття коду функціональними тестами Розробити додаткові тести, щоб покрити не покриті раніше частини коду Виконати структурне тестування програмного забезпечення Документувати результати тестів у Звіті про тестування програмного забезпечення (Software Test Report) Перевірити погодженість Software Test Plan, Software Test Specification, Software Test Report, а також їх відповідність вимогам Software Design (потрібна пряме трасування вимоги) Усунути або дозволити аномалії *Примітка. Для генерації виконання, і документування тестів, для аналізу результатів тестів, а також для аналізу тестового покриття повинні використовуватися комп'ютерні інструменти*	Виконати тестування функцій ІБ

Таблиця 2.9.

Фаза Fault Insertion Testing

Найменування фази	Виконувані завдання	Забезпечення функціональної безпеки	Забезпечення інформаційної безпеки
Тестування «засівом» дефектів (Fault Insertion Testing)	Інтеграція програмного забезпечення й апаратних засобів Тестування шляхом внесення несправностей в апаратне й програмне забезпечення для перевірки функції самодіагностики	Підготувати План тестування «засівбою» дефектів (Fault Insertion Test Plan) на підставі результатів FMECA у частині діагностичного покриття (формальне трасування вимог не потрібно) Підготувати Специфікацію тестування «засівбою» дефектів (Fault Insertion Test Specification) шляхом розробки тест кейсів для кожної тестованої вимоги Виконати тестування «засівбою» дефектів Документувати результати тестів у Звіті про тестування «засівбою» дефектів (Fault Insertion Test Report) Перевірити погодженість Fault Insertion Test Specification, Fault Insertion Test Report, а також їх відповідність вимогам Fault Insertion Test Plan (потрібна пряме трасування вимоги) Усунути або дозволити аномалії *Примітка. Для генерації виконання, і документування тестів, а також для аналізу результатів тестів повинні використовуватися комп'ютерні інструменти*	Протестувати функції виявлення атак і перекладу системи в безпечний стан у випадку виявлення атаки

Таблиця 2.10.

Фаза Integration Testing

Найменування фази	Виконувані завдання	Забезпечення функціональної безпеки	Забезпечення інформаційної безпеки
Інтеграційне тестування (Integration Testing)	Функціональне тестування інтегрованих апаратних і програмних компонентів на відповідність вимогам Проекту (SAD)	Підготувати План інтеграційного тестування (Integration Test Plan) на основі трасування вимог Проекту (SAD) Підготувати Специфікацію інтеграційного тестування (Integration Test Specification) шляхом розробки тест кейсів для кожної тестованої вимоги Виконати інтеграційне тестування Документувати результати тестів у Звіті про інтеграційне тестування (Integration Test Report) Перевірити погодженість Integration Test Plan, Integration Test Specification, Integration Test Report, а також їх відповідність вимогам SAD (потрібна пряме трасування вимоги) Усунути або дозволити аномалії *Примітка. Для генерації виконання, і документування тестів, а також для аналізу результатів тестів повинні використовуватися комп'ютерні інструменти*	

Таблиця 2.11.

Фаза Validation Testing

Найменування фази	Виконувані завдання	Забезпечення функціональної безпеки	Забезпечення інформаційної безпеки
Валідаційне тестування (Validation Testing)	Інтеграція системи Функціональне тестування інтегрованої системи на відповідність вимогам Специфікації (SRS)	Підготувати План валідаційного тестування (Validation Test Plan) на основі трасування вимог Специфікації (SRS) Підготувати Специфікацію валідаційного тестування (Validation Test Specification) шляхом розробки тест кейсів для кожної тестованої вимоги Виконати валідаційне тестування Документувати результати тестів у Звіті про валідаційне тестування (Validation Test Report) Перевірити погодженість Validation Test Plan, Validation Test Specification, Validation Test Report, а також їх відповідність вимогам SRS (потрібна пряме трасування вимоги) Усунути або дозволити аномалії *Примітка. Для генерації виконання, і документування тестів, а також для аналізу результатів тестів повинні використовуватися комп'ютерні інструменти*	

Трасування вимог є одним з процесів більш широкої області знань, званої інженерія вимог (Requirements Engineering).

Під вимогою розуміється документальне уявлення умови або властивості, якому повинна відповідати система або системний компонент, для того, щоб задовольняти контрактом, стандартам, специфікації або іншим формальним документам.

Трасування вимог є методом управління вимог, що змінюються і відносяться до них артефактів. Трасування вимог вирішує три основні завдання:

- забезпечує реалізацію на нижньому рівні всіх вимог верхнього рівня, - запобігає появі на нижньому рівні недокументованих функцій, - забезпечує тестування всіх вимог.

Розглянемо на прикладі, як виконується трасування (див. Рис. 2.8.). Приклад взятий з проекту по сертифікації контролера RadICS. Для виконання трасування вимог документи повинні бути підготовлені до цього процесу шляхом розстановки ідентифікаторів вимог і тегів, що визначають межі формулювань вимог. Для цього наведемо фрагмент Safety Requirements Specification (рис.2.8).

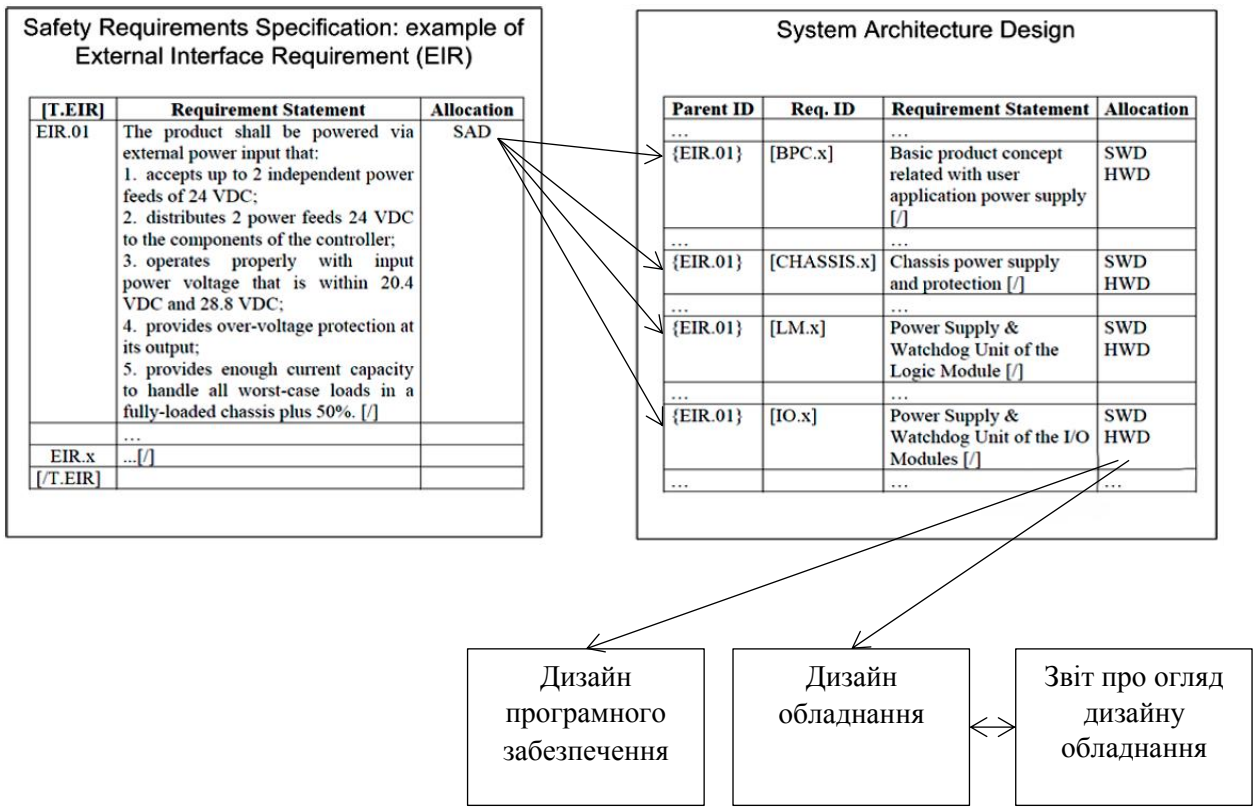


Рис. 2.8. Приклад трасування вимог до вимог програмного забезпечення

Документ розроблений в табличному вигляді. Для кожного з вимог введений ідентифікатор, в даному випадку це EIR.01, вимоги до інтерфейсу електроживлення. Сканування документа для трасування виконується інструментальним засобом DEUS компанії exida. DEUS є макрос MS Office. Початок вимоги визначається ідентифікатором, а закінчення вимога символом [/]. В поле Allocation вказується документ нижнього рівня, куди трасуються вимоги специфікації. В даному випадку це System Architecture Design. В System Architecture Design також повинна бути виконана розмітка вимог тегами. Крім того, для кожного з вимог повинен бути зазначений його джерело в документі верхнього рівня, так зване батьківське вимога. Для нашого вимоги вийшло відношення один-ко-многим, тобто вимога до електроживлення відобразилося в вимоги і до базової концепції продукту, і до шасі, і до всіх модулів. Далі кожну з вимог SAD трасується в Hardware Design & Software Design. Остаточні результати трасування можуть бути представлені у вигляді набору матриць, що відображають співвідношення між вимогами документів. Зворотній трасування вимог виконується від документів

нижнього рівня до документів верхнього рівня щоб переконатися в тому, що в продукті не з'явився зайвий недокументований функціонал.

При тестуванні трасування здійснюється шляхом вилучення вимог з проектних документів. Для складних проектів розробка документів з тестування може проходити в два етапи. Спочатку розробляється план тестування, що містить перелік тестованих вимог, а потім в специфікації тестування для кожного з вимог розробляються тест кейси. При тестуванні здійснюється пряма трасування вимог від проектного документа до плану і специфікації тестування, а потім до звіту по тестуванню. Для тестування методом засіву дефектів перелік тестів витягується зі звіту за FMECA шляхом аналізу діагностованих відмов. Для обгрунтованого безлічі відмов складається набір тестів, на яких перевіряються функції діагностики. Зворотній трасування тут не критична, оскільки, якщо будуть проводитися додаткові тести, не обумовлені проектними документами, то це не вплине на безпеку.

Висновки. В даному розділі розглянули вимоги до організації життєвого циклу систем управління (АСУ ТП, вбудовані системи, інтернет речей); запропонована єдина структура процесів, що підтримують виконання вимог як до інформаційної, так і до функціональної безпеки. Для комп'ютерних систем управління основні ризики відносяться до порушень властивості ФБ. Методи забезпечення ФБ спрямовані на захист від випадкових відмов апаратних засобів, викликаних фізичним старінням елементів, а також на захист від систематичних відмов, викликаних недосконалістю процесів проектування.

Дотримання вимог ГОСТ 15.005 при виготовленні комплектів ПТК для АЕС дозволяє істотно зменшити витрати на розробку ІКС, що автоматично означає підвищення економічної ефективності виробництва. Виготовлення тиражованих одиничних виробів за чинним ТЗ є економічно вигідним як для постачальника, так і для замовника так як ТЗА поставляються до замовника повністю зібраними, що не вимагає витрат на оплату додаткових складальних операцій, що позначається на економії коштів замовника.

Розділ 3. МОДЕЛІ ТА МЕТОДИ ОРГАНІЗАЦІЇ ВЕРИФІКАЦІЇ БІЗНЕС КРИТИЧНИХ ПРОГРАМНИХ СИСТЕМ УПРАВЛІННЯ ЕКОНОМІЧНИМ ОБ'ЄКТОМ

3.1. Модель та методи життєвого циклу верифікації бізнес критичних програмних систем управління економічним об'єктом

При виконанні сертифікації на відповідність вимогам МЕК 61508 до того або іншого рівня SIL (Safety Integrity Level), важливо продемонструвати, що, по-перше, у продукт закладений необхідний і достатній механізм забезпечення функціональної безпеки (ФБ), а, по-друге, що необхідний і достатній набір методів забезпечення ФБ застосовувався в процесі розробки. Виходячи із цього, умовно розділимо методи забезпечення ФБ на технічні й організаційні.

Методи забезпечення ФБ спрямовані на захист від випадкових відмов апаратних засобів, викликаних фізичним старінням елементів, а також на захист від систематичних відмов, викликаних недосконалістю процесів проектування. Оскільки в сучасному світі ФБ немислимо розглядати у відриві від інформаційної безпеки (ІБ), то методи забезпечення ФБ також, здебільшого, забезпечують і захист від кібер атак.

Опис методів забезпечення ФБ утримується в МЕК 61508-7 «Методи і засоби». Однак, щоб розібратися у вимогах до застосування того або іншого методу, однієї тільки частини 7 недостатньо.

У частині 2 МЕК 61508, яка присвячена забезпеченню ФБ систем і апаратних засобів, і в частині 3, що містить вимоги до ПО, є три істотні додатки, які визначають методи й засобу по захисту від випадкових і систематичних відмов. Структура МЕК 61508 така, що детально методи й засобу описано в частині 7. Таким чином, вимоги до захисту від відмов із частин 2 і 3 МЕК 61508 трасуються в описи методів і засобів по забезпеченню ФБ у частині 7.

Додаток А частини 2 МЕК 61508 розглядає контроль відмов апаратних засобів у ході експлуатації, причому мається на увазі, що це можуть бути як

випадкові, так і систематичні відмови. Детальний опис методів захисту від випадкових відмов утримується в Додатку А частини 7 МЕК 61508. Методи захисту від випадкових відмов розділені на категорії, залежно від типу розглянутих апаратних засобів, наприклад, електричні компоненти, електронні компоненти, процесорні модулі, пам'ять і т.д.

Додаток В частини 2 МЕК 61508 розглядає методи захисту від систематичних відмов апаратних засобів протягом життєвого циклу. Слід зазначити, що МЕК 61508 розглядає в якості джерел систематичних відмов апаратних засобів помилки проектування, помилки експлуатації й зовнішні екстремальні впливи (кліматичні, механічні, радіаційні й інші). Детальний опис відповідних методів утримується в Додатку В частині 7 МЕК 61508. Структура викладу методів захисту від систематичних відмов апаратних засобів має свої особливості, наприклад, у категорію B.1 General measures and techniques входять такі різні методи, як Project Management, Documentation, Separation, Diversity. Далі методи розподілені по етапах життєвого циклу.

Додаток А частини 3 МЕК 61508 містить посібник з вибору методів розробки й тестування програмного забезпечення з метою досягнення повноти безпеки.

Детальний опис відповідних методів міститься в додатку Із частини 7 МЕК 61508. Тут у категорії C.1 General взагалі не описуються ніякі методи. Далі методи розподілені по категоріях: вимоги й детальна дизайн, архітектурний дизайн, інструментальні засоби й мови програмування, верифікація й модифікація, оцінювання ФБ.

Щоб проілюструвати виклад вищесказаного, розглянемо найпростішу таблицю, яка визначає методи, застосовувані для забезпечення функціональної безпеки програмного забезпечення на етапі розробки специфікації вимог

Допустимо, нас цікавить рівень повноти безпеки SIL3. Ті методи, які в графові «SIL3» позначені, як HR (Highly Recommended), повинні обов'язково застосовуватися. У всякому разі органу, що сертифікує, буде практично неможливо пояснити, чому не використовувався той або інший обов'язково рекомендований метод. Методи, позначені, як R (Recommended), можуть застосовуватися, однак, від

них також можна обґрунтовано відмовитися. З Таблицею А1 (SIL3) тут усе досить просто. Рекомендується застосування формальних методів, однак, пріоритет застосування напівформальних методів вище (Highly Recommended). Тому, якщо застосовуються напівформальні методи, то від формальних можна відмовитися. Застосування прямого і зворотного трасування також є обов'язковим. Застосування програмного забезпечення для підтримки трасування й розробки специфікації є очевидним.

Як ми вже говорили, методи забезпечення безпеки доцільно розділити на організаційні й технічні. Розглянемо тепер ці дві групи.

Організаційні методи забезпечення ФБ

протягом усього життєвого циклу.

Житєвий цикл ІБ і ФБ

Реалізація життєвого циклу ІБ і ФБ є вимогою, у МЕК 61508 особливий акцент робиться на такі аспекти, як:

— структурований процес розробки системи та програмного забезпечення;

— реалізація процесу верифікації й валідації, що полягає в поетапному виконанні оглядів, аналізу й тестування;

— супровід продукту після релізу з урахуванням зворотного зв'язку за результатами експлуатації.

Використання кращих практик і стандартів кодування

МЕК 61508 вимагає впроваджувати безпечне використання мов програмування, яке полягає в застосуванні мов зі строгою типізацією й підтримкою структурного програмування, при цьому рекомендується використовувати обмежену безліч конструкцій мови. Наприклад, для мікроконтролерів систем безпеки перевага віддається мові С, а не С++. Для визначення правил кодування й заборонених конструкцій використовуються відповідні стандарти, наприклад MISRA.

Стандарти кодування й кращі практики визначають ряд угод (coding conventions), які використовуються при розробці ПО в якості вимог до коду. Такі

угоди містять у собі правила найменування й коментування, правила відступів і оформлення коду, обмеження по складності і т.д.

Розповсюдженою практикою є так зване захисне програмування, коли при виникненні якої-небудь критичної проблеми програмне забезпечення завершує роботу заздалегідь передбаченим образом, тобто переводить систему в безпечний стан.

Використання сертифікованих компіляторів і бібліотек

При розробці програмного забезпечення, важливого для безпеки, необхідно гарантувати, що компілятор перетворить вихідний програмний код у здійсненний прогнозованим і детермінованим образом. Для цього використовуються сертифіковані компілятори й транслятори коду, а також сертифіковані бібліотеки програмних компонентів. В останні кілька років провідні виробники мікропроцесорів і ПЛІС випустили версії компіляторів і супутніх бібліотек, які були сертифіковані й, відповідно, можуть бути використані для розробки ПО систем безпеки. Сучасні тенденції в інженерії електроніки приводять до все більшої інтеграції дизайну програмного й апаратного забезпечення програмувальних компонентів, тому роль і вплив на безпеку інструментарію розробки буде усе більше зростати.

Контроль якості при виробництві апаратних засобів

При виробництві апаратних засобів, особлива увага приділяється якості друкованих плат із установленими на них електронними компонентами. Контроль якості складається з таких складових, як:

— розробка й верифікація проекту апаратних засобів;

— керування якістю закуповуваних матеріалів і компонентів;

— керування виробництвом;

— інспекція якості на виробничій ділянці;

— тестування випущених апаратних засобів.

Використання формальних і напівформальних нотацій

Ще одним організаційним методом, необхідним МЕК 61508, є застосування формальні й напівформальних нотацій для розробки специфікації вимог і проектів

систем, а також програмних і апаратних компонентів. У даний час найпоширенішими напівформальними нотаціями для проектування програмувальних систем є IDEF і UML.

Для програмувальних логічних контролерів розроблені й описані в стандарті МЕК 61131-3 типові мови програмування, підтримувані більшістю середовищ розробки. Найпоширенішим є графічна мова FBD (Function Block Diagram), яка фактично є формальною нотацією для проектування програмного забезпечення.

Технічні методи забезпечення ФБ

Почнемо з розгляду архітектури автоматизованих систем управління технологічним процесом (АСУ ТП). До складу системи входять:

— компоненти електропостачання;

— польове встаткування (датчики й виконавчі механізми);

— програмувальні логічні контролери, включаючи модулі введення й виводу й керуючі модулі;

— мережне устаткування, сервера й компонента людино-машинного інтерфейсу.

Резервування

Тепер розглянемо, яким чином для АСУ ТП може бути реалізоване резервування (redundancy). Резервування може реалізовуватися як для окремих компонентів або їх груп, так і для системи в цілому. Саме такий випадок пропонується вашій увазі (див. Рис. 3.1.).

Почнемо з електроживлення. В ідеалі, максимальна незалежність забезпечується при електроживленні незалежних каналів системи від незалежних джерел. На схемі показане, що перший канал харчується від джерела змінного струму, а другий канал – від постійного струму. Тоді, при проблемах з живленням в одній із систем енергопостачання буде знеструмлюватися тільки один з каналів.

Забезпечення безперервності і якості електроживлення навіть в екстремальних умовах є життєво важливим аспектом забезпечення безпеки систем керування.

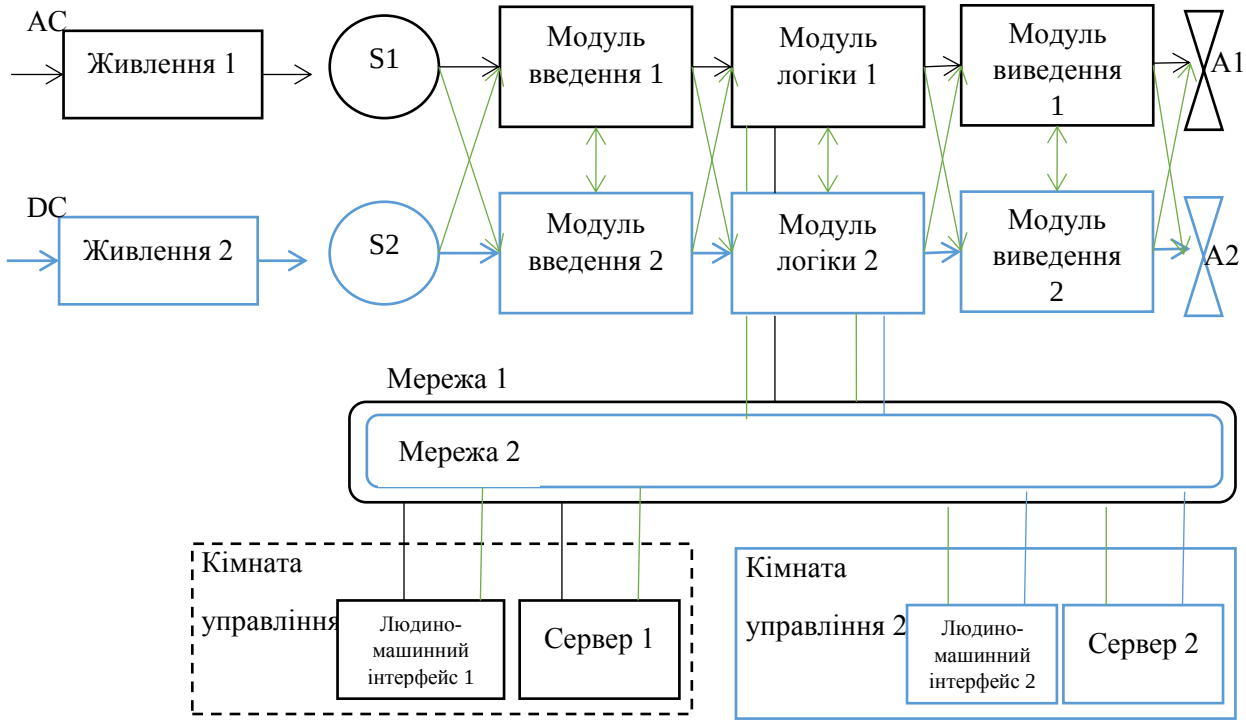


Рис. 3.1. Дублювання компонентів автоматизованої системи управління технологічних процесів

Можуть застосовуватися резервовані датчики, контролери й виконавчі механізми. Між каналами можуть бути організовані протоколи інформаційного обміну (вони позначені на схемі зеленим кольором) або ж може бути реалізована максимальна незалежність між каналами, і тоді обміну не буде.

Крім того, може бути реалізована дубльована мережна архітектура й дубльований людино-машинний інтерфейс із дубльованими обчислювальними компонентами й сховищами даних.

Різноманітність (диверсійність) при резервуванні

Диверсійністю (divercity) називається вид резервування, коли в резервних каналах та сама функція виконується різними шляхами, наприклад, із застосуванням різного встаткування або різного програмного забезпечення.

Звичайне резервування не захищає від систематичних відмов, викликаних помилками проектування. Тому, якщо версії системи спроектовані по-різному, та кількість загальних систематичних відмов каналів (так званих відмов по загальній причині) знизиться (у всякому разі, теоретично знизиться). Це враховується за

допомогою так званого β-фактора, який показує відношення кількості відмов (або інтенсивності відмов) по загальній причині до загальної кількості відмов (або інтенсивності відмов).β-фактор залежить від застосовуваної стратегії диверсійності. Чим більше відмінність між каналами, тем нижче значення β-фактора (див. Рис. 3.2.).

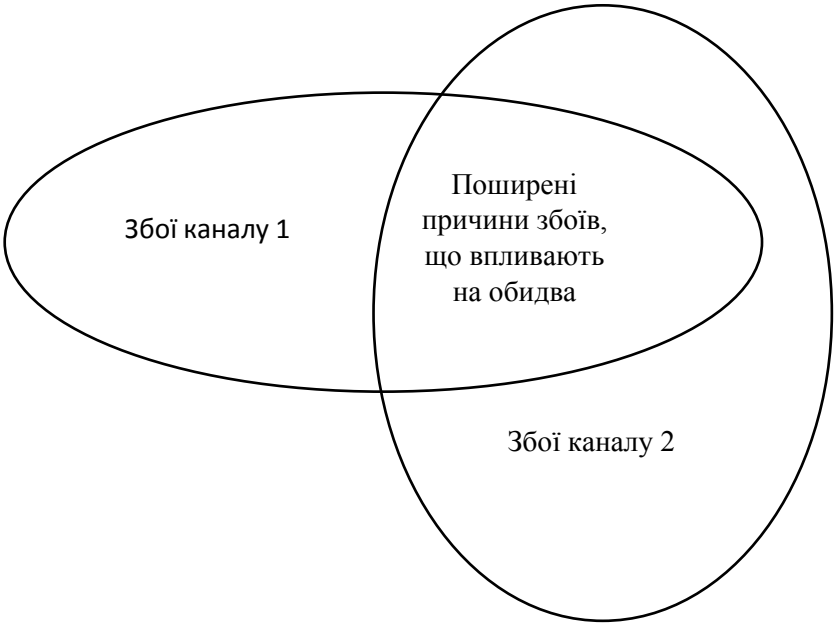


Рис. 3.2. Зниження кількості відмов по загальній причині при використанні різних (диверсійних) дубльованих каналів (джерело: МЕК 61508)

Звичайно, застосування диверсійності є вкрай дорогим методом, що підвищують вартість системи в рази, однак у деяких галузях з високими ризиками, наприклад, в атомній енергетиці, це є обґрунтованим і потрібним стандартом.

Незалежність і поділ компонентів

Ще одним методом, що доповнюють резервування, є принцип незалежності й поділу (independency and separation) компонентів, спрямований на запобігання поширення відмов між системами і їх компонентами. Поділ може бути фізичним, коли, наприклад, канали системи фізично перебувають у різних приміщеннях або на значній відстані друг від друга (див. Рис. 3.3). Застосовується також електричний поділ, що включає, наприклад, гальванічну ізоляцію, а також функціональна незалежність і незалежність комунікацій, наприклад, екранування кабелів і поділ

електричних і сигнальних кабелів.

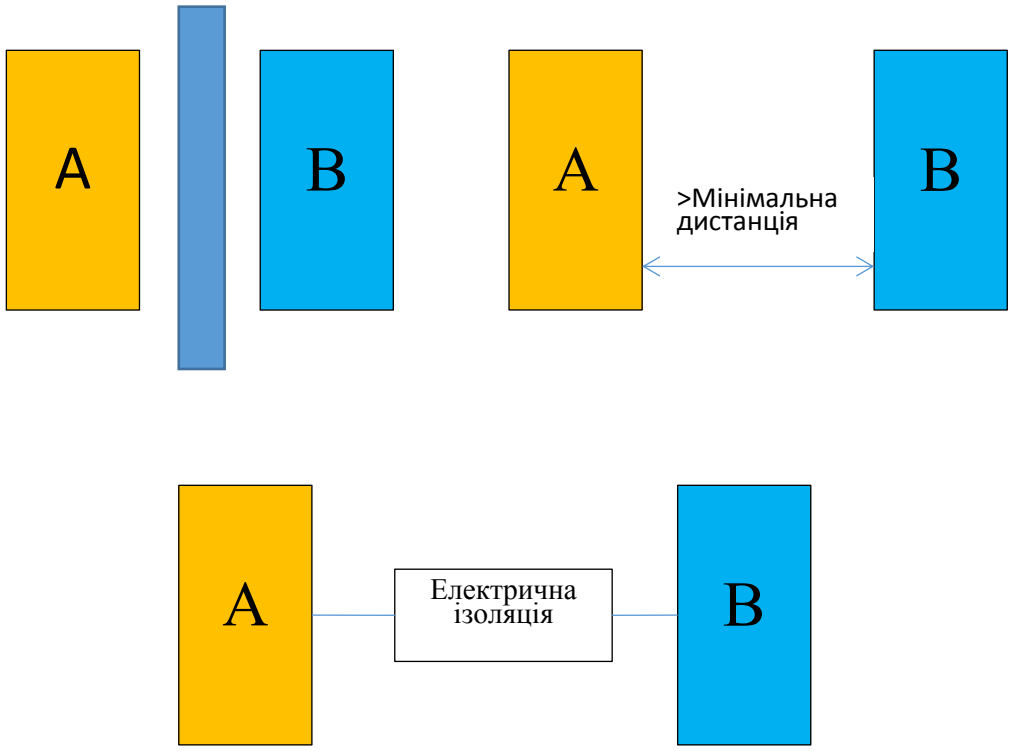


Рис. 3.3. Фізична й електрична незалежність каналів (джерело: МЕК 60709)

Самодіагностика

Самодіагностику цифрових пристроїв спрощено можна описати в такий спосіб (див. Рис. 3.4).

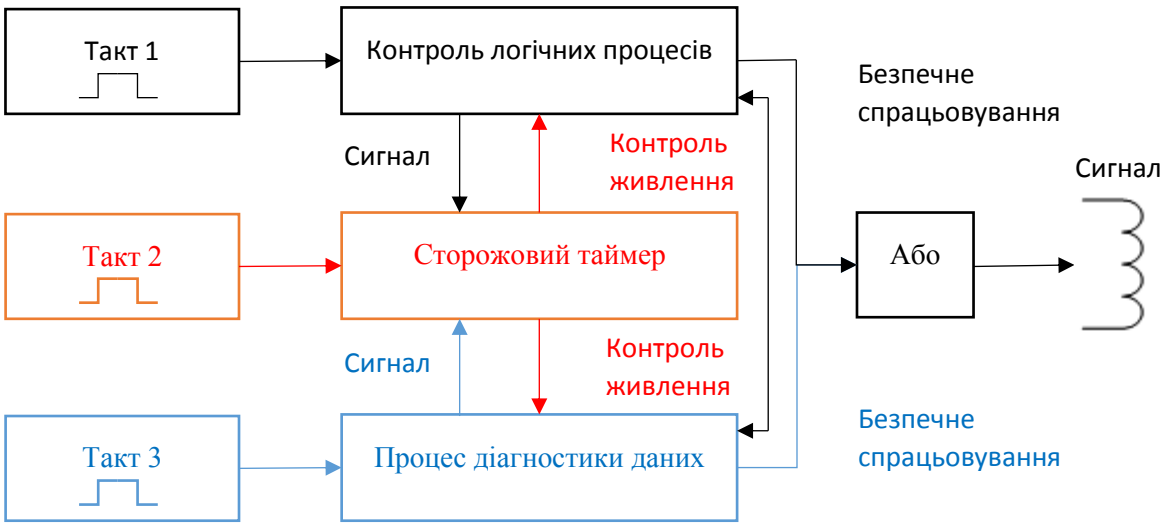


Рис. 3.4. Реалізація діагностики в системах керування

Поряд з основними алгоритмами цифрового керування, паралельно в системі реалізується обробка діагностичних даних і сторожовий таймер (watchdog). Усі ці три процеси виконуються незалежно один від одного, можуть використовуватися незалежні джерела тактової частоти, різні мікросхеми і т.д.

Watchdog контролює найпростіший відгук від мікросхем, що виконують обробку даних, і при виявленні проблеми (припиненні відгуку) відключає живлення й переводить систему в безпечний стан. Крім того, ватчдог може контролювати рівень живлення й видавати аналогічну команду на відключення при небезпечному відхиленні живлення від заданого рівня. Безпечний стан для систем безпеки, як правило, полягає в знятті живлення з вихідних аналогових і дискретних виходів. При необхідності, система безпеки може подавати живлення на виконавчі механізми, але тоді на виході потрібні додаткові перетворювачі сигналу.

Якщо самодіагностика виявила критичну проблему (наприклад, відмова апаратних вузлів, порушення конфігурації апаратних або програмних засобів, порушення передачі даних і т.п.), то видається команда на переклад системи в безпечний стан, яка виконується так само, як якби команда зробила від основної керуючої логіки.

Тепер узагальнимо, які типові функції повинна виконувати самодіагностика цифрових пристроїв.

Важливою функцією діагностики є контроль конфігурації програмних і апаратних засобів. Ця властивість впливає також на забезпечення інформаційної безпеки. У процесі функціонування кожний апаратний модуль періодично передає інформацію про свій серійний номер і про конфігурацію завантаженого ПО (наприклад, чек-суму). У випадку порушення конфігурації система виконує задані захисні дії, аж до переходу в безпечний стан і відключення живлення.

Ще одним варіантом виконання контролю зависань є внутрішні або зовнішні таймери, що контролюють час виконання окремих модулів керуючої логіки. Завдання можуть перезапускатися кілька раз, у випадку декількох невдалих перезапусків також може ухвалюватися розв'язок про перехід у безпечний стан.

Важливою функцією систем керування є забезпечення точності виміру вхідних і вихідних аналогових сигналів. Для діагностування точності вимірів можуть застосовуватися резервовані АЦП і ЦАП, у яких рівняються результати обробки й видається діагностичне повідомлення про збіг або розбіжність результатів.

Велика увага в системах керування приділяється передачі пакетів даних, як по комунікаційних каналах, так і при обробці, розподіленої між компонентами програмного й апаратного забезпечення. Тут для діагностування застосовуються такі методи, як підтвердження передачі, контроль таймаутів, контроль цілісності й послідовності передачі пакетів даних, циклічні коди (CRC). Для захисту інформації при передачі даних можуть застосовуватися алгоритми шифрування.

Захист від впливів навколишнього середовища

Ще одна група методів забезпечення безпеки спрямована на захист від несприятливих впливів. Для забезпечення функціонування систем керування застосовується вентиляція й кондиціювання повітря, проектують конструкції, стійкі до вібрації й іншим механічним впливам, застосовуються системи пожежогасіння й негорючі матеріали, матеріали й покриття, стійкі до хімічних і радіаційних впливів.

Серйозна увага приділяється забезпеченню електромагнітної сумісності. Для цього здійснюється фільтрація й придушення електромагнітних перешкод різного роду, як зовнішніх, так і власних, для обмеження впливу на іншу апаратуру.

Спеціальні стандарти ATEX застосовуються для конструювання вибухобезпечних систем. Стандарти IP застосовуються для конструювання систем, захищених від впливу пили й вологи.

Захист від помилок персоналу

Персонал, що здійснює експлуатацію систем керування, може, як знизити, так і побільшати ризики. Багато техногенних аварій були обумовлені людським фактором. У той же час, відомо багато випадків, коли професійні дії дозволяли уникати катастроф і загибелі людей. Тому, розробка людини-машинного

інтерфейсу з урахуванням вимог ергономічності й захисту від помилок оператора також є важливим напрямком забезпечення безпеки.

Особливості забезпечення інформаційної безпеки АСУ ТП

При розгляді методів, спрямованих на забезпечення ФБ, важливо також забезпечити ІБ. У МЕК 61508 на цей рахунок утримується лише кілька загальних слів. Ми зупинимося на декількох особливостях АСУ ТП, які визначають підхід до забезпечення ІБ таких об'єктів. Такими уразливостями є: периметр мережі, вилучені підключення, фаєрволи устаткування, що здобувається, змінні носії інформації (у першу чергу, що використовують Usb-Порти), а також програмне забезпечення контролерів і комунікаційні лінії.

Сегментування мережі

Основою забезпечення ІБ для АСУ ТП є сегментування мережі й зонування розміщення встаткування.

Рекомендується реалізовувати в АСУ ТП, як мінімум, одну демілітаризовану зону (DMZ), що розділяє корпоративну й керуючу локальні мережі.

Контроль доступу

Ще однієї особливістю забезпечення ІБ АСУ ТП є активне використання контролю доступу, який може бути реалізовано декількома способами.

Шафи з устаткуванням забезпечуються замками й контактними датчиками відкриття дверей, які видають сигнали на табло сигналізації.

Моніторинг і збереження великого обсягу діагностичних даних, у тому числі, пов'язаних з ІБ, надає широкі аналітичні можливості.

Оскільки, комунікаційні лінії є вразливими компонентами, доступ до них також контролюється, як на фізичному, так і на логічному рівні. Використання комунікаційних ліній і портів повинне бути обґрунтовано обмежене. Для систем безпеки потрібне використання тільки односпрямованих комунікацій з використанням фірмових протоколів, що відрізняються від широко розповсюджених промислових протоколів.

Для зміни настроювань ПО й самого ПО, а також для зміни конфігурації апаратних засобів повинна виконуватися спеціальна авторизація.

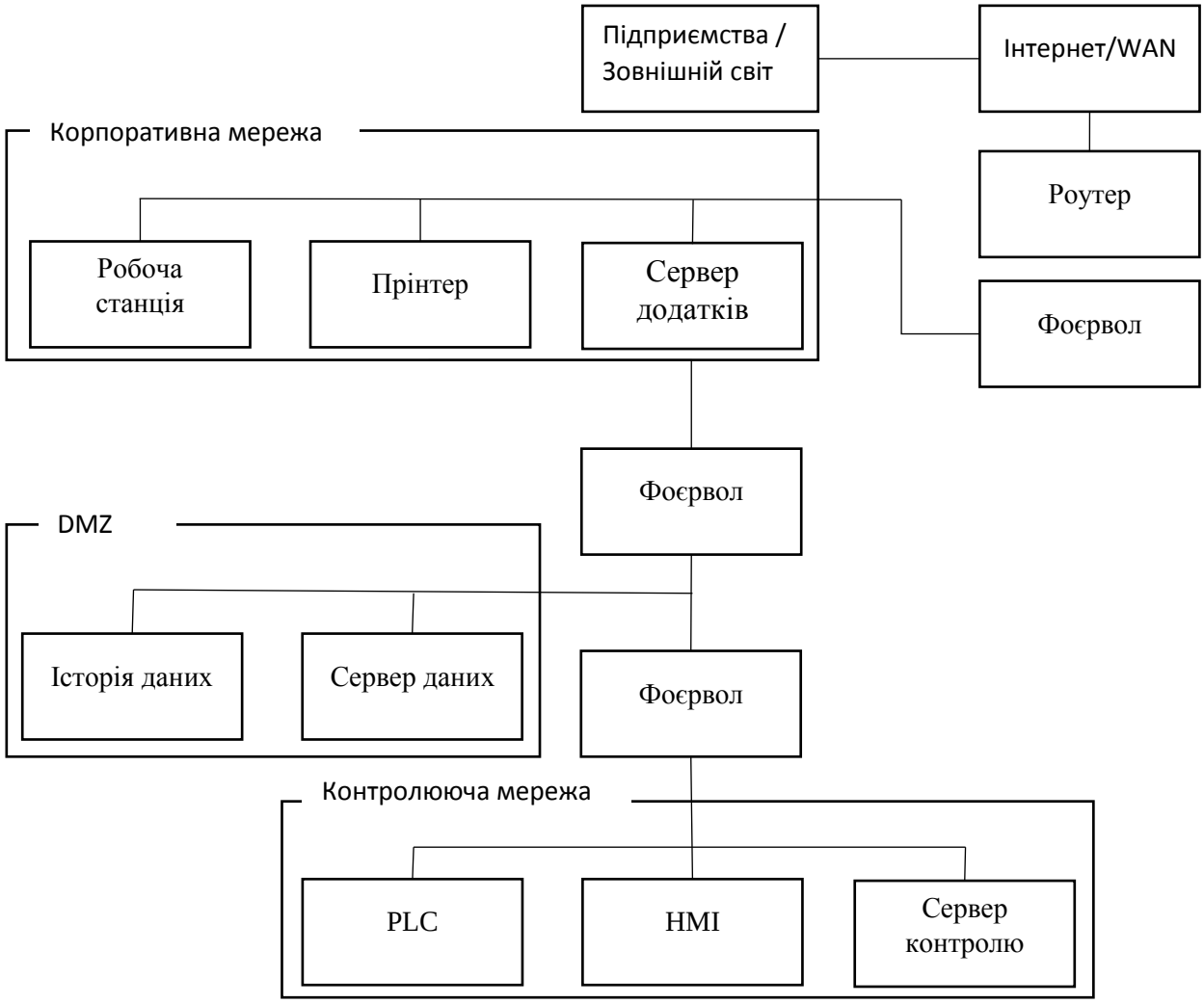


Рис. 3.5. Структура мережі АСУ ТП із DMZ (джерело: NIST 800-82)

Висновки

При реалізації життєвого циклу ІБ та ФБ важливими є такі аспекти:

— структурований процес розробки систем та програмного забезпечення;

— реалізація процесу верифікації й валідації, що полягає в поетапному виконанні оглядів, аналізу й тестування;

— супровід продукту після релізу з урахуванням зворотного зв'язку за результатами експлуатації.

Важливою функцією діагностики є контроль конфігурації програмних і апаратних засобів. Ця властивість впливає також на забезпечення інформаційної безпеки. Моніторинг і збереження великого обсягу діагностичних даних, у тому числі, пов'язаних з ІБ, надає широкі аналітичні можливості.

3.2. Модель та методичні підходи аналізу та управління ризиками у життєвому циклі верифікації бізнес критичних систем управління та їх програмного забезпечення

Для того, щоб зробити аналіз ризиків, потрібен ще один крок в розгляді стандарту МЕК 61508.

Справа в тому, що функціональна безпека – це досить формалізована властивість, оскільки системи, важливі для безпеки, є предметом державного ліцензування у всіх країнах.

Стандарт оперує терміном електрична/ електронна/ програмувальна електронна (Е/Е/ПЕ) система (electrical/electronic/programmable electronic).

Особливістю стандарту є ризик-орієнтований підхід. Залежно від ризику, який техногенний об'єкт створює для навколишнього середовища, життя й здоров'я людей, установлюються ризики для відмов систем керування.

Наприклад, на системи захисту атомних реакторів. Для них у режимі постійної роботи відмови повинні відбуватися не частіше, чим один раз в 1000 років експлуатації (10 мільйонів годин наробітку на відмову). Такі показники задаються не для одиничного об'єкта, а для «флоту», тобто для безлічі однотипних об'єктів. Начебто б, відмови є досить рідкими подіями, адже жодна атомна електростанція не проробить тисячу років. Однак, якщо врахувати, що у світі експлуатується більш 400 атомних реакторів, то для «флоту» ми вже одержимо цифру одна відмова в 2,5 року, що звучить набагато критичніше. Під час Чорнобильської й Фукусімської ядерних катастроф системи аварійного захисту не спрацювали так, як очікувалося проектувальниками. Це ще один аргумент на користь важливості розгляду функціональної безпеки.

Для зниження значень ризиків нижче заданих показників реалізується комплекс організаційно-технічних заходів, які також регламентовані в МЕК 61508, залежно від припустимої величини ризику відмови.

Крім того, МЕК 61508 являє собою верхній рівень цілого сімейства галузевих стандартів, які деталізують вимоги до функціональної безпеки для систем керування медичним устаткуванням, автомобільним і залізничного транспортом, АСУ ТП і т.д.

Перша редакція МЕК 61508 була розроблено з 1998 по 2000 роки.

Друга редакція МЕК 61508 випущена в 2010.

Стандарт поділений на частини, які легко зрозуміти по їхніх назвах.

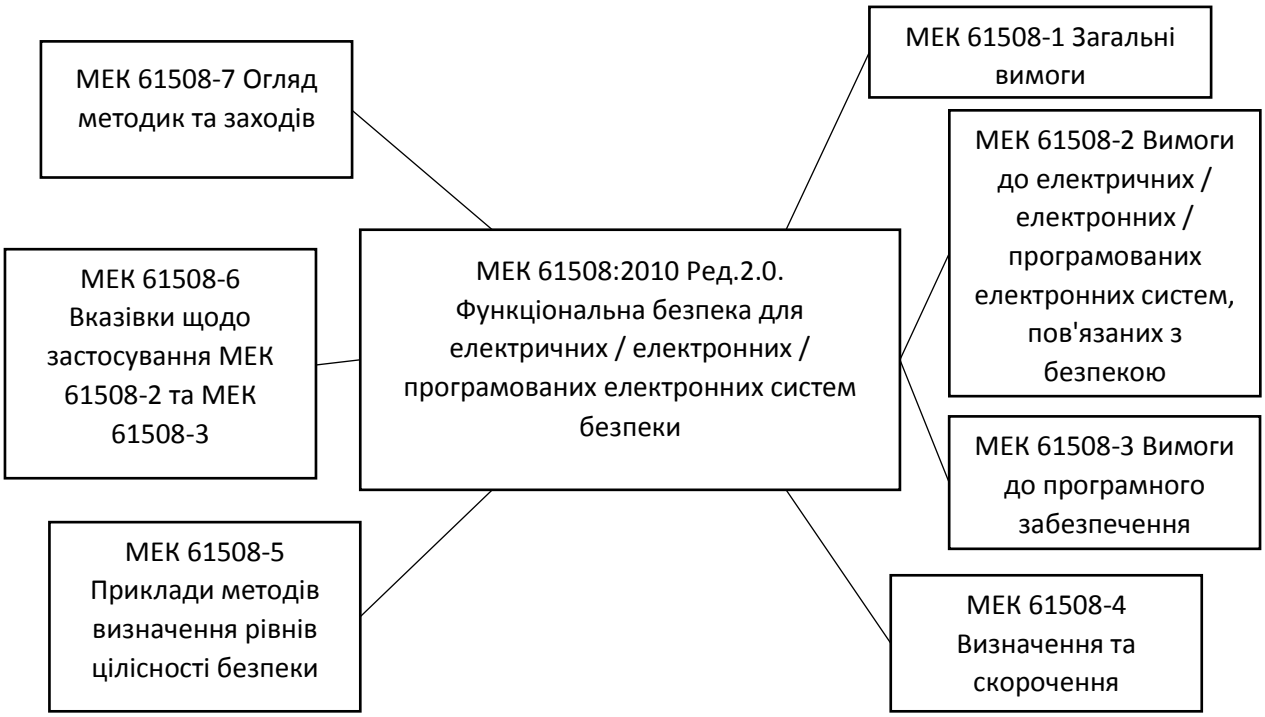


Рис. 3.6. Стандарт МЕК 61508

Звичайно, між частинами МЕК 61508 існують досить складні зв'язки, що відбито на Рис. 3.6.самого стандарту.

Термінологія МЕК 61508: базові терміни по безпеці

Далі терміни вибірково цитуються згідно з українськомовним текстом МЕК 61508-4, а потім дається їхня авторська інтерпретація.

При спробі зв'язати всі сутності в єдине ціле, в результаті вийшла наступна схема Рис. 3.7. Важливо відзначити, що комп'ютерна система керування (КСУ) є лише однією з багатьох ланок по зниженню ризиків. Існує безліч так званих пасивних заходів захисту, наприклад, ремінь безпеки в автомобілі або в літаку.

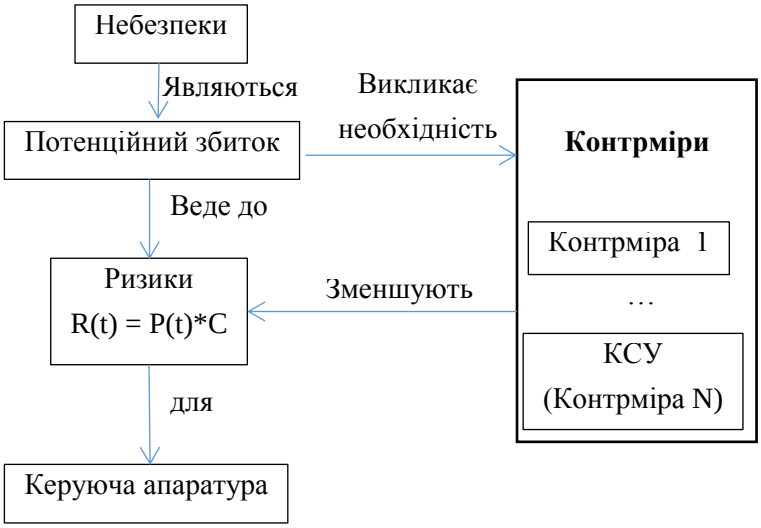


Рис. 3.7. Зв’язки між комп'ютерними системами керування

Ризик є показником безпеки, і на понятті ризику необхідно буде звернути увагу надалі, при розгляді цих самих показників. Поки приведемо простий приклад. Ходіння людей по краю даху. Імовірність падіння з даху за інших рівних умов не залежить від висоти будинку. А от ступінь збитку залежить. Тоді й ризик падіння з даху 10-поверхового будинку буде вище, ніж ризик падіння з даху одноповерхового будинку. А от ризик падіння з даху 10-поверхового будинку практично дорівнює ризику падіння з даху 100-поверхового будинку, оскільки, якщо ймовірність падіння однакова, те й наслідки (збиток) падіння тут, на жаль, також однакові.

Цікавим є поняття припустимого (прийнятного) ризику. Воно залежить від історичного й гуманітарного контексту. Чи правда, що найвищими цінностями сучасного світу є людське життя та піклування про навколишнє середовище , яке формує та підтримує це саме життя? Реальний стан техногенних об'єктів демонструє, наскільки держава й суспільство реалізують проголошені цінності.

Ще одним принциповим поняттям є «безпечний стан». Наприклад, одна з найважливіших систем безпеки, система протиаварійного захисту (ПАЗ), повинна зупинити функціонування керованого об'єкта. Як це відбувається? Як правило, шляхом розриву електричних кіл (це вже залежить від технологічних алгоритмів керування встаткуванням), що відбувається шляхом перекладу вихідних дискретних сигналів у стан «логічний 0» (так званий принцип de-energize to trip, щоб система могла відробити й при аварійній втраті електропостачання). При необхідності, «логічний 0» може бути інвертовано в «логічну 1» через проміжні реле.

Термінологія МЕК 61508: терміни, що ставляться до функцій безпеки й повноті безпеки (safety integrity)

В попередніх розділах було згадано, що поняття функціональної безпеки містить у собі реалізовані функції безпеки й повноту (інтегрованість) виконання цих функцій.

З погляду визначення, що приводиться, повнота безпеки (safety integrity), ця властивість фактично зводиться до безвідмовного виконання функцій безпеки, тобто розглядається, як частина безвідмовності, яка, у свою чергу, є складової класичної надійності. Насправді, з інших положень МЕК 61508 випливає, що повнота безпеки є більш складною властивістю, пов'язаним з такими атрибутами, як ремонтопридатність, готовність, довговічність, інформаційна безпека. Термінологічні й таксономічні аспекти складових надійності й безпеки являють собою суміжну галузь знань.

Ще одним центральним поняттям у МЕК 61508 є рівень повноти безпеки (Safety Integrity Level, SIL). Значення SIL установлює залежно від того, наскільки вплив керованого устаткування створює ризик для людей і навколишнього середовища.

Виходячи із цього, установлений ризик відмови й для самої комп'ютерної системи керування. Наприклад, раніше було вказано, що для системи захисту атомного реактора наробіток на відмову повинна становити не менш, чим 10 мільйонів годин. Це відповідає SIL3. Взагалі, прийнято вважати, що SIL4 можуть

відповідати лише найбільш прості пристрої. Для програмувальних логічних контролерів (ПЛК), використовуваних в АСУ ТП, досяжним є SIL3.

Зі структури визначень також випливає, що повнота безпеки ділиться на дві складові: повнота безпеки, що стосується систематичних відмов (сюди ж попадає повнота безпеки програмного забезпечення) і повнота безпеки апаратних засобів.

Перша складова вимагає застосовувати заходу захисту від систематичних відмов, викликаних помилками проектування. Для цього необхідно вдосконалювати процеси проектування й розробки, тестування, керування конфігурацією, проектного менеджменту й т.п. Це віддалено нагадує рівні Capability Maturity Model Integration (CMMI), але прямо до них не трасується. Для кожного зі значень SIL визначений набір методів захисту від систематичних відмов, причому їх кількість і «строгість» зростає з підвищенням SIL.

Повнота безпеки апаратних засобів пов'язана із захистом від випадкових відмов і забезпечується застосуванням компонентів з високим рівнем безвідмовності й самодіагностики, і, звичайно ж, резервуванням.

Можна досягти рівня SIL2 при одноканальній конфігурації ПЛК. Тоді резервована конфігурація дасть SIL3. При цьому процеси розробки (systematic capability), повинні відповідати SIL3.

Тепер, за аналогією з попереднім розділом, спробуємо застосувати структуру оточення (небезпека, збиток, ризики, контрміри, кероване устаткування) для комп'ютерної системи керування (КСУ). Тут мова йде про небезпеки для виконання функцій безпеки КСУ, оскільки їх невиконання пов'язане з ризиком. Для зниження цього ризику застосовуються різні заходи щодо забезпечення повноти безпеки. Як ми вже знаємо, ці заходи спрямовані на захист від випадкових і систематичних відмов Рис. 3.8. Використаємо отриману схему зі схемою з попереднього розміру і таким чином отримаємо дворівневу структуру (Рис. 3.9), що демонструє термінологічне середовище функціональної безпеки.

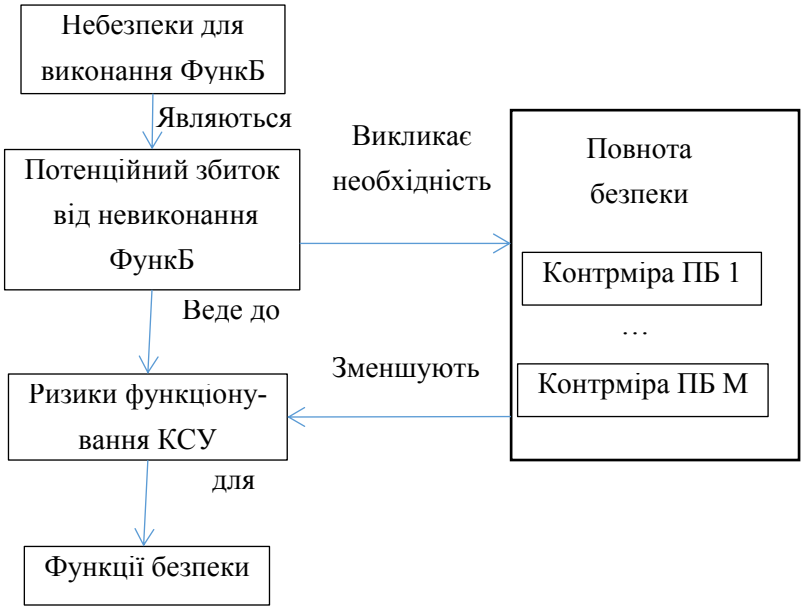


Рис. 3.8. Захист від випадкових і систематичних відмов програмного забезпечення

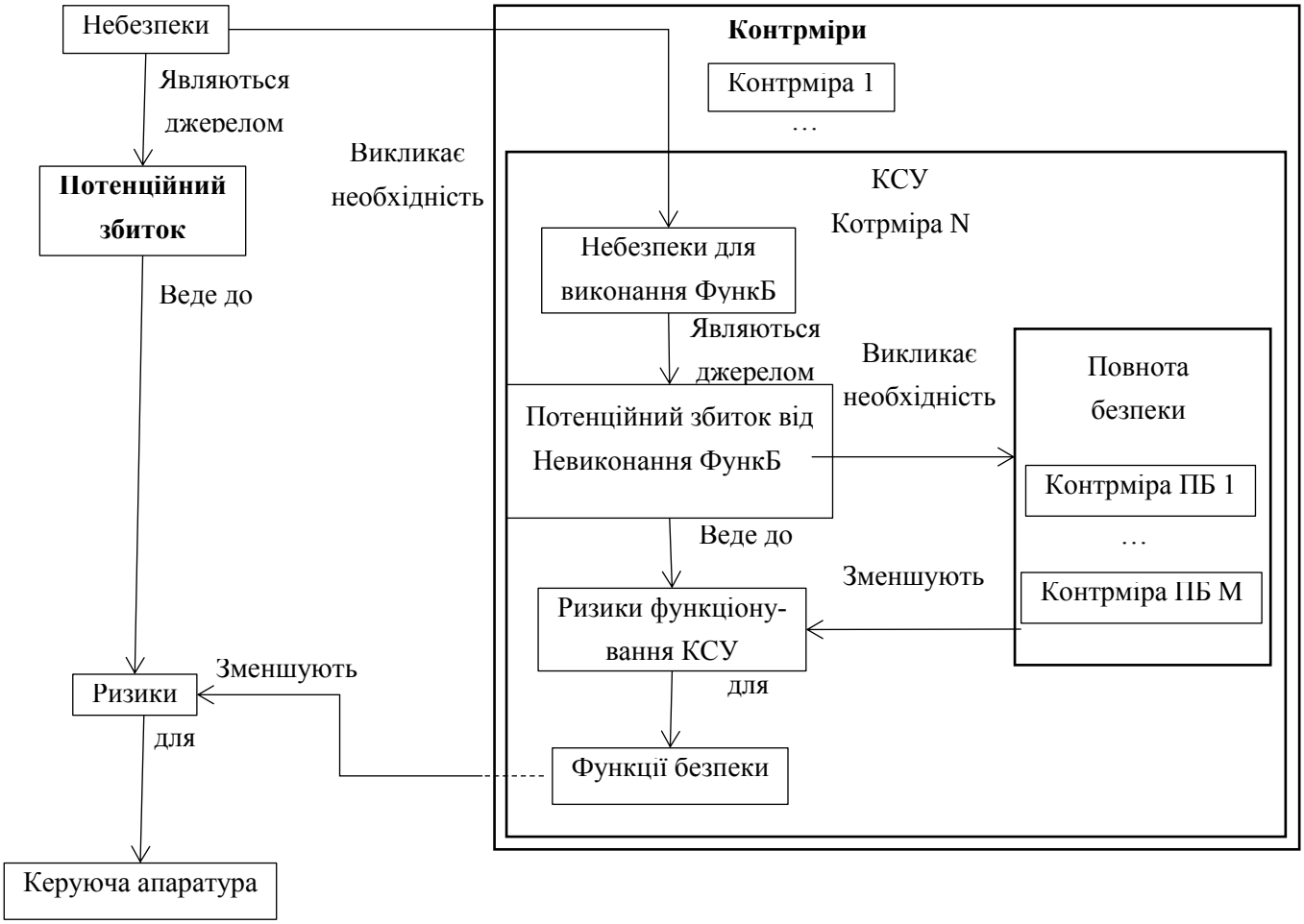


Рис. 3.9. Дворівнева структура захисту від випадкових і систематичних відмов програмного забезпечення

Висновки. Комп'ютерна система керування є лише одним з багатьох заходів по зниженню ризиків. Повнота безпеки апаратних засобів пов'язана із захистом від випадкових відмов і забезпечується застосуванням компонентів з високим рівнем безвідмовності й самодіагностики, і, звичайно ж, резервуванням.

3.3. Планування та ефективність використання автоматизації процесів верифікації бізнес критичних систем управління економічним об’єктом

Розглянемо шляхи удосконалення роботи інформаційних комп’ютерних систем управління АЕС за допомогою бібліотек OSVVM , UVVM та формальної верифікаціїї.

Для початку вияснемо, що таке бібліотеки OSVVM , UVVM та формальної верифікаціїї.

Методологія перевірки VHDL з відкритим вихідним кодом (Open Source VHDL Verification Methodology (OSVVM)): перевірка рівня VHDL ASIC, досить проста для ПЛІС

OSVVM пропонує:

-Моделювання На Рівні Транзакцій

-Генерація Random тесту

-Функціональне покриття з гачками для інтеграції бази даних покриття UCIS

-Інтелектуальне покриття генерація випадкових тестів

-Утиліти для синхронізації процесів testbench

-Файли розшифрування

-Ведення журналу помилок і звітів-оповіщення й підтвердження

-Фільтрація повідомлень

-Модель пам'яті

OSVVM реалізований у вигляді бібліотеки з безкоштовним відкритим вихідним кодом. OSVVM використовує ці пакети для створення функцій, які конкурують із реалізаціями на основі мови як у лаконічності, простоті, так і в можливостях. Зокрема, OSVVM використовує ці пакети для створення інтелектуальної методології перевірки покриття, яка є кроком уперед у порівнянні з іншими методологіями перевірки, такими як UVM Systemverilog.

OSVVM можна використовувати VHDL testbench, частково або повністю в міру необхідності. Це дозволяє змішувати нашу сигнатурну методологію “інтелектуального покриття " з іншими методологіями перевірки, такими як спрямована, алгоритмічна, файлова й обмежена випадкова. Не викидайте існуючі тестові середовища VHDL або їх моделі, а використовуйте їх повторно.

Поговоримо більш детальніше про можливості в OSVVM.

Моделі рівня транзакцій (transaction-level) реалізовані , як сутності і архітектури. Для підключення transaction-level до середовища тестування використовують записи. Щоб дозволити одному запису реалізувати інтерфейс

транзакції, ми використовуємо функції resoluton в Resolutionpkg OSVVM. Resolutionpkg був уперше випущений в 2016.11 OSVVM релізі, і в майбутньому на ньому буде презентовано більше блоків.

Генерація випадкового тесту(Random test)

RandomPkg OSVVM надає набір утиліт для рандомізації значення в зазначеному діапазоні.

Усе випадкове тестування засноване на рівномірній рандомізації. Рівномірна рандомізація повторює значення зі швидкістю log N, де N-Число генеруємих значень.

В OSVVM ми використовуємо інтелектуальну рандомізацію покриття у якості нашої основної методології рандомізації, щоб уникнути повторного стимулу, і обмежену випадковість як методологію уточнення в наших тестах.

Функціональне Покриття(Coverage)

Функціональне покриття-це код, який спостерігає за виконанням плану тестування. Таким чином, щоб відслідковувати, чи були виконані важливі умови, набори значень або послідовності значень, відповідні до вимог до дизайну або інтерфейсу, функцій або граничних умов.

Функціональне покриття має важливе значення для будь-якого підходу до створення Random test, оскільки це єдиний спосіб визначити, що зробив тест. У міру того як складність конструкції збільшує, охоплення 100% функціональний переконує нас, що всі деталі в плані випробувань були випробувані. Значення в 100% показує нам, що було повне покриття коду, і це вказує на те, що тестування виконане.

Інтелектуальна Методологія Рандомізації Coverage

Перевірка починається із плану тестування, який визначає всі елементи в конструкції, які необхідно протестувати. OSVVM, як і інші передові методології, використовує функціональне покриття для спостереження умов на інтерфейсах і в рамках конструкції для перевірки того, що елементи, процеси в плані тестування, відбулися. Таким чином, функціональне покриття допомагає визначити, коли виконується тестування.

На відміну від інших методологій, у методології інтелектуального покриття OSVVM функціональне покриття є основною директивою – це те, де ми починаємо наш процес. Інтелектуальне покриття виконується в наступних кроках:

-Написані високоточного функціонального покриття (FC)

-Уточнює початкову рандомізацію за допомогою послідовного коду

Ключовим моментом інтелектуального покриття є те, що ми рандомізуємо, використовуючи функціональне покриття. Потім, при необхідності, ми уточнюємо рандомизацию з використанням послідовного коду й будь-якого методу генерації послідовності, включаючи обмежені випадкові, алгоритмічні, спрямовані або методи читання файлів.

Утиліти для синхронізації процесів testbench. Пакет OSVVM, Tbutilpkg, надає утиліти testbench для синхронізації процесів, а також утиліти для генерації і скидання процесів. Tbutilpkg був уперше випущений в 2016.11 OSVVM release, і в майбутньому на ньому буде презентовано більше блогів.

Файли розшифрування. Можливість розшифрування OSVVM спрощує розробку різних частин тестового середовища. Для цього він надає внутрішній ідентифікатор файлу (Transcriptfile) і підпрограми для відкриття (Transcriptopen) файлів, закриття (Transcriptclose) файлів, виводу (print and writeline) і перевірки, чи відкриття файлів (Istranscriptopen).

Модель пам'яті (Memorypkg) спрощує процес створення ефективних структур даних для моделей пам'яті. Memorypkg уперше був випущений в 2016.11 OSVVM release, і в майбутньому на ньому буде презентовано більше блоків.

OSVVM є безкоштовними та працює на звичайних Vhdl-Симуляторах (таких як Modelsim і Questasim від Mentor, а також Active-hdl від Aldec і Riviera-pro) без додаткових ліцензій. Єдина необхідна підтримка спеціальних мов-це захищені типи VHDL-2002 і тип integer_vector VHDL-2008 .

UVVM (Universal VHDL Verification Methodology)-це безкоштовна й відкрита методологія й бібліотека для створення дуже структурованих testbenches на основі VHDL.

Огляд, читаність, ремонтопридатність, розширюваність і повторне використання-усе це життєво важливо для ефективності і якості розробки FPGA. Uvvm VVC (VHDL Verification Component) Framework була випущена в 2016 році для обробки саме цих аспектів.

Для початку. Зверніть увагу, що дана бібліотека має два різні рівні складності. VVC Framework і Vvcs для середніх і просунутих testbenches, а також службова бібліотека й Bfms для простого використання - і як основу для більш просунутих testbenches.

UVVM у цей час складається з наступних елементів:

- Библиотеки Утилит;
- Framework VVC (компонент провірки VHDL) - включаючи службову бібліотеку;
- Bfms (Bus Functional Models)
- Vvcs, який буде використовуватися з Uvvm VVC Framework і може бути об'єднаний з Bfms

UVVM -Це система компонентів верифікації, яка дозволяє реалізувати дуже структуровану архітектуру testbench для обробки будь-якої складності верифікації - від дійсно простій до дійсно складної. Ключовою перевагою цієї системи є дуже простий програмно-подібний VHDL test sequencer, який може управляти всією вашою архітектурою testbench з будь-якою кількістю компонентів перевірки. Це вимагає огляду, читабельності й ремонтопридатності до нового рівня.

Великою перевагою тут-це унікальний огляд, читаність, ремонтопридатність, розширюваність і повторне використання, які ви одержуєте від наявності кращої можливої архітектури testbench. Ще однією важливою перевагою тут є те, що люба кількість команд може бути видана від тестового sequencer - таким чином, дозволяючи повний контроль. Це дає відмінний контроль над testbench і Vvcs.

Ви можете, звичайно, об'єднати UVVM з будь-якими іншими застарілими або сторонніми testbench або моделями перевірки.

Верифіка́ція форма́льна — в інформаційних технологіях, доказ, або заперечення відповідності системи у відношенні до певної формальної

специфікації або характеристики, із використанням формальних методів математики.

Тестування програмного забезпечення не може довести, що система, алгоритм або програма не містить ніяких помилок і дефектів та задовольняє певним властивостям. Це може зробити формальна верифікація.

Формальна верифікація може використовуватися для перевірки таких систем, як програмне забезпечення, представлене у вигляді вихідних текстів, криптографічні протоколи, комбінаторні логічні схеми, цифрові схеми з внутрішньою пам'яттю.

Верифікація являє собою формальний доказ на абстрактній математичній моделі системи, в припущенні про те, що відповідність між математичною моделлю і природою системи вважається заданим. Наприклад, щодо побудованої моделі або математичного аналізу, доказ правильності алгоритмів і програм.

Доказ може бути автоматизований повністю лише для дуже невеликого кола простих теорій, тому важливого значення набуває його автоматична перевірка і для цього приведення до належного вигляду.

Для підтримки строгості при перевірці доказу верифікатором слід перевірити ще й верифікатор, для чого потрібен ще один верифікатор і так далі. Отриманий нескінченний ланцюг верифікаторів можна було б згорнути, побудувавши веріфікуючий себе верифікатор, що володіє здатністю розвернутися до застосовного на практиці. Серед великої кількості мов, що використовуються при проведенні формальної верифікації, було обрано Property Specification Language (PSL). PSL використовується для опису властивостей, які потребують підтримки в проекті під верифікацію та надає засоби для написання специфікацій, які легко читаються і математично точні. Це призначення використовується для функціональної специфікації, з одного боку, і як вхід в програми функціональної верифікації, з іншого. Таким чином, специфікація PSL - це виконувана специфікація для апаратного проекту.

Після проведення аналізу методів верифікації ми можемо порівняти і вияснити, який метод буде більш ефективнішим. Для вирішення завдання було обрано п’ять альтернативних методів верифікації VHDL проектів (Табл. 3.1)

Таблиця 3.1

Альтернативні методи верифікації VHDL проектів

№ п/п	Метод верифікації	Позначення
1	Базовий набір бібліотек	БНБ
2	Open Source VHDL Verification Methodology	OSVVM
3	Universal VHDL Verification Methodology	UVVM
4	Формальна верифікація на мові PSL	PSL
5	Open Source VHDL Verification Methodology + Universal VHDL Verification Methodology	OSVVM+UVVM
6	Open Source VHDL Verification Methodology + Universal VHDL Verification Methodology+ Формальна верифікація на мові PSL	OSVVM+UVVM+PSL

Визначимо наступні критерії для оцінки методів верифікації з урахуванням загальноприйнятої практики та стандартів:

- функціональність (functionality) - здатність вирішувати потрібний набір задач, видавати потрібні результати, здатність до взаємодії, відповідність стандартам і правилам, відповідність ПЗ наявним індустріальним стандартам, нормативним і законодавчим актам, іншим регулюючим нормам, здатність запобігати неавторизованому доступу до даних і програм;
- час виходу на ринок (time to market) – час, за який компанія долає шлях від розробки концепції до першого відвантаження товару;
- зручність використання (usability) - здатність ПЗ бути зручним у навчанні та використанні, а також привабливим для користувачів;

- зручність супроводу (maintainability) - зручність проведення всіх видів діяльності, пов'язаних із супроводом програм;

Числові оцінки критеріїв для методів верифікації VHDL проектів будемо проводити, базуючись на аналізі даних з сайтів виробників відновідних систем, на експертних оцінках провідних фіхівців в визначеній галузі, а також на досвіді їх впровадження в визначеній предметній області (Табл. 3.2).

Таблиця 3.2

Порівняльна характеристика методів верифікації програмного забезпечення

Показник	**БНБ**	**OSVVM**	**UVVM**	**PSL**	**OSVVM + UVVM**	**OSVVM + UVVM+PSL**
Зручність використання	Зручний і простий	Більш складна але зручна	Більш складна але зручна	Більш складна (потрібно переписувати вимоги)	Більш складна але зручна	Складна
Час виходу на ринок	Найбільший	Середній	Середній	Середній	Середньо-великий	Середньо-великий
Зручність супроводу	Повна підтримка	Досить повна підтримка	Досить повна підтримка	Досить повна підтримка	Досить повна підтримка	Досить повна підтримка
Функціональність	-Легко навчати, читати і писати	-Легко навчати, читати і писати -Короткий синтаксис -Бібліотеки з додатковими можливостями -Рандомізація -Синхронізація процесів	-Легко навчати, читати і писати -Короткий синтаксис -Бібліотеки з додатковими можливостями -Повний контроль над сигналами	-Короткий синтаксис -Строго певна формальна семантика	-Легко навчати, читати і писати -Короткий синтаксис -Бібліотеки з додатковими можливостями -Рандомізація -Синхронізація процесів -Повний контроль над сигналами	-Короткий синтаксис -Бібліотеки з додатковими можливостями -Рандомізація -Синхронізація процесів -Повний контроль над сигналами -Строго певна формальна семантика

Звичайно ж, методи верифікації відрізняються не тільки наведеними параметрами. Було відібрано лише ті критерії, які дійсно можуть якось вплинути

на наш вибір. Додаткові критерії можуть бути й іншими. Їх кількість також може відрізнятися від обраної нами. Слід враховувати, що:

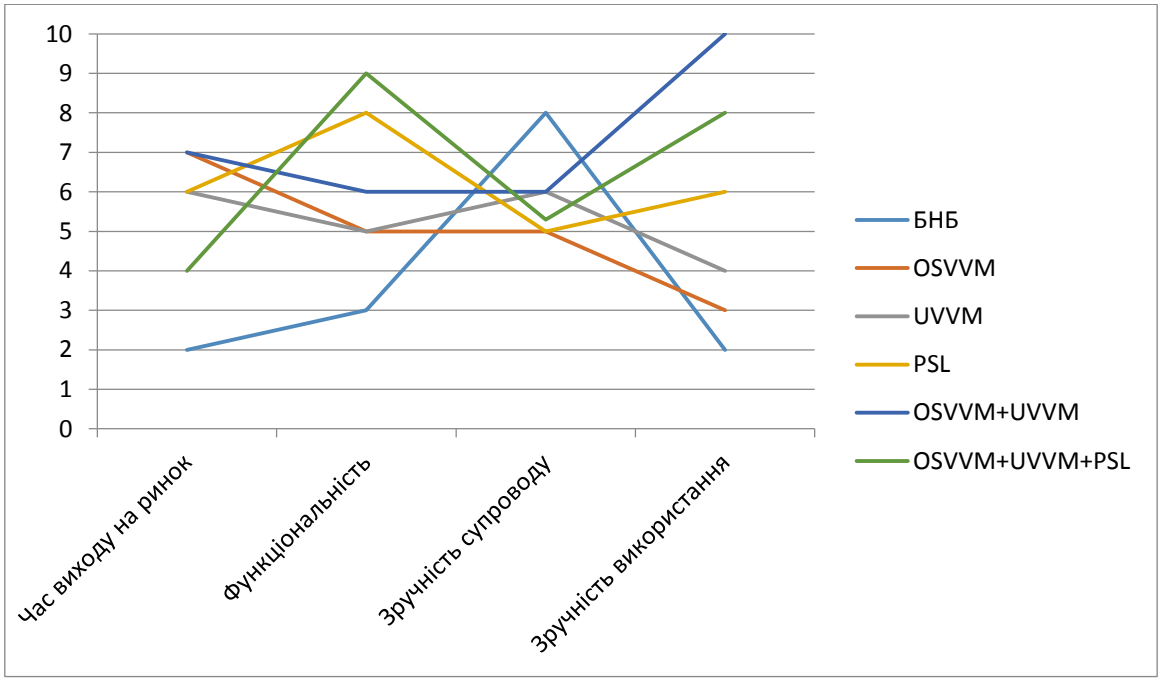
- повинна бути можливість зібрати інформацію за кожним додатковим критерієм для всіх відібраних альтернатив;
- кількість критеріїв не повинно перевищувати 4-5, щоб не збільшити трудомісткість обробки даних до нерозумних меж.

Вся підготовча робота проведена. Тепер на практиці застосуємо метод діаграм для вибору методу верифікації. Першим кроком буде оцінка критеріїв.

Почнемо з того, що представимо критерії, за якими порівнювалися бібліотеки, за 10 бальною шкалою.

Таблиця 3.3

Порівняльна характеристика методів верифікації



Як бачимо з діаграми (Таблиця 3.3), базовий набір бібліотек не дає достатніх можливостей для якісної верифікації критичного програмного забезпечення.

Таблиця .3.4

Результати порівняльної характеристики методів верифікації

	БНБ	OSVVM	UVVM	PSL	OSVVM+UVVM	OSVVM+UVVM+ PSL
Час виходу на ринок	2	7	6	6	7	4
Функціональність	3	5	5	8	6	9
Зручність супроводу	8	5	6	5	6	5,3
Зручність використання	2	3	4	6	10	8
Разом	3,75	5	5,25	6,25	7,25	6,575

Як бачимо на Таблиця 3.4, бібліотеки Open Source VHDL Verification Methodology + Universal VHDL Verification Methodology дають можливість суттєво покращити швидкість роботи (в 2 рази) та якість верифікації критичного програмного забезпечення.

Висновки. Проведено огляд деяких популярних методів верифікації ПЗ FPGA. Розглянуто шляхи вдосконалення верифікації ПЗ інформаційних комп'ютерних систем управління АЕС за допомогою бібліотек OSVVM , UVVM та формальної верифікації. В результаті роботи дійшли висновку:

-використання бібліотек OSVVM , UVVM та формальної верифікації дає найбільш ефективний результат, спрощує верифікацію програмного забезпечення і прискорює цей процес більш, як у 2 рази.

Ключовою перевагою UVVM системи є дуже простий програмно-подібний VHDL test sequencer, який може управляти всією вашою архітектурою testbench з будь-якою кількістю компонентів перевірки.

OSVVM є безкоштовним та працює на звичайних Vhdl-симуляторах (таких як Modelsim і Questasim від Mentor, а також Active-hdl від Aldec і Riviera-pro) без додаткових ліцензій. Єдина необхідна підтримка спеціальних мов-це захищені типи VHDL-2002 і тип integer_vector VHDL-2008.

ВИСНОВКИ

У цій роботі були розглянуті питання верифікації й атестації ПЗ. Було доведено, що це дуже складні кроки в розробці будь-якого продукту, що вимагають уваги від фахівців найвищої кваліфікації, а від організації – великих вкладень. Але якими б вартісними не були ці процеси, економічна вигода від їхнього використання очевидна, адже система без збоїв не наносить збитків. Варто пам'ятати, що аварійні ситуації – рідкі події (особливо в КС), тому практично неможливо змоделювати їх під час тестування системи. Було встановлено, що вимоги безпеки ніколи не виключають ненадійну поведінку системи. За допомогою тестування й інших процесів атестації неможливо цілком довести відповідність системи вимогам безпеки.

В даний час здобуває велике значення оцінка захищеності систем, оскільки частіше за все системи поєднуються за допомогою мережі Інтернет. Вимоги захищеності в деяких відносинах подібні вимогам безпеки. Зокрема, вони визначають позаштатне поводження системи, а не її «робоче» поводження. Однак, як правило, неможливо визначити це поводження у виді простих обмежень, контрольованих системою.

Кінцевим користувачам дуже складно перевірити захищеність системи. Тому в Європі вироблені системи критеріїв оцінки захищеності, що контролюються експертами. Постачальники готового ПЗ можуть надати на розгляд свої кінцеві продукти для оцінки і сертифікації за різними критеріями захищеності. Верифікація й атестація повинні стати обов'язковими кроками в розробці ПЗ, нехай навіть найпростішого. Організації повинні також враховувати економічний стан на ринку ПЗ, бажання користувачів (уже було відзначено, що вимоги користувачів до ПЗ зростають). У сучасних умовах організації роботи підприємства все більше місця займає підвищення автоматизації виробництва та використання програмних технологій, що дозволяє підвищити рівень економічного розвитку підприємства.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. IEEE 829-2008. Standard for Software Test Documentation.
2. IEEE 1012-2012. Standard for Software Verification and Validation.
3. IEEE 1016-2009. IEEE Standard for Information technology – Systems Design - Software Design Descriptions.
4. IEEE 1028-2008. IEEE Standard for Software Reviews and Audits.
5. IEEE 1076-2008. VHDL Language Reference Manual.
6. IEC 61508-2010. Functional safety of electrical/ electronic/ programmable electronic safety-related systems – Part 3: Software requirements.
7. IEC 61508-2010. Functional safety of electrical/ electronic/ programmable electronic safety-related systems – Part 7: Overview of techniques and measure
8. ГОСТ 15.005-86 Система разработки и постановки продукции на производство
9. ГОСТ 19.001-77. Единая система программной документации (ЕСПД). Общие положения.
10. ГОСТ 19.003-80. ЕСПД. Схемы алгоритмов и программ. Обозначения условные графические.
11. ГОСТ 19.005-85. ЕСПД. Р-схемы алгоритмов и программ. Обозначения условные графические и правила выполнения.
12. ГОСТ 19.101-77. ЕСПД. Виды программ и программных документов.
13. ГОСТ 19.102-77. ЕСПД. Стадии разработки.
14. ГОСТ 19.103-77. ЕСПД. Обозначение программ и программных документов.
15. ГОСТ 19.104-78. ЕСПД. Основные надписи.
16. ГОСТ 19.105-78. ЕСПД. Общие требования к программным документам.
17. ГОСТ 19.106-78. ЕСПД. Требования к программным документам, выполненным печатным способом.
18. ГОСТ 19.201-78. ЕСПД. Техническое задание. Требования к содержанию и оформлению.

19. ГОСТ 19.202-78. ЕСПД. Спецификация. Требования к содержанию и оформлению.
20. ГОСТ 19.301-79. ЕСПД. Программа и методика испытаний. Требования к содержанию и оформлению.
21. ГОСТ 19.401-78. ЕСПД. Текст программы. Требования к содержанию и оформлению.
22. ГОСТ 19.402-78. ЕСПД. Описание программы.
23. ГОСТ 19.403-79. ЕСПД. Ведомость держателей подлинников.
24. ГОСТ 19.404-79. ЕСПД. Пояснительная записка. Требования к содержанию и оформлению.
25. ГОСТ 19.501-78. ЕСПД. Формуляр. Требования к содержанию и оформлению.
26. ГОСТ 19.502-78. ЕСПД. Описание применения. Требования к содержанию и оформлению.
27. ГОСТ 19.503-79. ЕСПД. Руководство системного программиста. Требования к содержанию и оформлению.
28. ГОСТ 19.504-79. ЕСПД. Руководство программиста. Требования к содержанию и оформлению.
29. ГОСТ 19.506-79. ЕСПД. Описание языка. Требования к содержанию и оформлению.
30. ГОСТ 19.601-78. ЕСПД. Общие правила дублирования, учета и хранения.
31. ГОСТ 19.602-78. ЕСПД. Правила дублирования, учета и хранения программных документов, выполненных печатным способом.
32. ГОСТ 19.603-78. ЕСПД. Общие правила внесения изменений.
33. ГОСТ 19.604-78. ЕСПД. Правила внесения изменений в программные документы, выполненные печатным способом.
34. ГОСТ 19.701-90 (ИСО 5807-85). ЕСПД. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения.
35. Закон України “Про метрологію та метрологічну діяльність”. – 1993. http://rada.gov.ua.

36. ДСТУ 2462-94. Сертифікація. Основні поняття. Терміни та визначення. - Чинний від 01.01.95. -К.: Держстандарт України, 1994. - 27 с.
37. ДСТУ 2853-94. Програмні засоби ЕОМ. Підготовлення і проведення випробувань. - Чинний від 01.01.96. -К.: Держстандарт України, 1994. - 17 с.
38. ДСТУ 2851-94. Програмні засоби ЕОМ. Документування результатів випробувань. - Чинний від 01.01.96. -К.: Держстандарт України, 1994. - 12 с.
39. ISO/IEC 12119. IT - Software packages - Quality requirements and testing, 1994. 29 p.
40. ISO/IEC 9126-1. Software engineering - Product quality - Part 1: Quality model, 2001. 26 p.
41. А. С. Алпєєв Верифікація та валідація програмованих керуючих систем АЕС. http://www.sstc.com.ua/documents/journal/2010/3/5_3_2010_text_ru.pdf
42. Береза А.М. Основи створення інформаційних систем: Навч. посібник. 2 видання,перероблене і доповнене – К.: КНЕУ, 2001.
43. Вовк О.Б. Аналіз та оцінювання якості програмного продукту (поняття, терміни, означення). http://ena.lp.edu.ua:8080/bitstream/ntb/8885/1/06.pdf
44. Георгіаді Н.Г. Моніторинг стану інформаційної системи управління економічним розвитком підприємства. http://vlp.com.ua/files/21_30.pdf
45. Гушко С.В., Шайкан А.В. Управлінські інформаційні системи: Навч. посібник. – Львів: Магнолія плюс, 2006. – 320 с.
46. Марченко Е. Что такое качество программного обеспечения. http://softwaretesting.ru/library/testing/general-testing
47. Карпенко М. Ю. Технології створення програмних продуктів та інформаційних систем : навч. посібник / М. Ю. Карпенко, Н. О. Манакова, І. О. Гавриленко ; Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова. – Харків : ХНУМГ ім. О. М. Бекетова, 2017. – 93 с.

48. Б.М.Конорев, В.В.Сергієнко, І.Б.Туркін Доказова незалежна верифікація та прогнозування прихованих дефектів критичного програмного забезпечення на базі диверсного вимірювання інваріантів // Інженерія програмного забезпечення №1(5) 2011

49. Контролювання та регулювання економічного розвитку підприємства: проблеми, методологічні та прикладні аспекти: Монографія / О.Є.Кузьмін, С.В.Князь, Н.О.Шпак, В.А.Новицький. – Львів: В-во НУЛП, 2006. – 148 с.

50. Коцовський В.М. Супровід програмних систем: Методичний посібник для студентів спеціальності "Інженерія програмного забезпечення" / В. М. Коцовський. — Ужгород: Видавництво УжНУ "Говерла", 2016. — 52 с.

51. Орлик С. Введение в программную инженерию и управление жизненным циклом программного обеспечения. Программная инженерия. Сопровождение программного обеспечения. http://sorlik.blogspot.com.

52. Райчев І.Е. Проблеми сертифікації програмного забезпечення автоматизованих систем контролю // Вісник НАУ. -2004. -№1. -С. 23-28.

53. Райчев І.Е., Харченко О.Г. Концепція побудови сертифікаційної моделі якості програмних систем // Проблемы программирования. -2006. -№2-3. - С. 275-281.

54. Райчев И.Э., Харченко А.Г., Яцков Н.А. Исследование методов тестирования программных модулей обработки полетной информации // Вісник КМУЦА. - 2000. - №1-2. - С.127-133.

55. Райчев И.Э., Харченко А.Г., Яцков Н.А. Методы создания тестовых наборов данных при сертификационных испытаниях комплексов программ контроля полетов // Вісник НАУ. - 2001. - №1. - С. 126-132.

56. Ястребенецький М.О., Розен Ю.В., Виноградська С.В. та ін. Безпека атомних станцій. Система керування й захисту ядерних реакторів. Російською мовою.- К: ТОВ «Основа Принт», 2011.-С. 32-128.

ДОДАТКИ

ДОДАТОК А

Матриці попарних порівнянь для критеріїв

Зручність супроводу

ПО Верифікації	Умовні одиниці
ModelSim	6
QuestaSim	5
Quartus	3
Формальна верифікація	1
ModelSim + OSVVM + UVVM	8

	ModelSim	QuestaSim	Quartus	Формальна верифікація	ModelSim + OSVVM + UVVM	Оцінки компонент власного вектора	Нормалізовані оцінки вектора пріоритету
ModelSim	1	2	7	9	2	3,021900	0,413406
QuestaSim	1/2	1	6	8	1	1,888175	0,258308
Quartus	1/7	1/6	1	1/2	1/7	0,279334	0,038214
Формальна верифікація	1/9	1/8	2	1	1/2	0,425142	0,058161
ModelSim + OSVVM + UVVM	1	1	7	2	1	1,695218	0,231911
Сума	2,7540	4,2917	23,0000	20,5000	4,6429	7,309769	

Відношення узгодженості (ВУ) =	**8,82%**	Повинно бути < 10%, допускається < 20%.

Зручність використання

ПО Верифікації	Умовні одиниці
ModelSim	5
QuestaSim	4
Quartus	3
Формальна верифікація	3
ModelSim + OSVVM + UVVM	5

	ModelSim	QuestaSim	Quartus	Формальна верифікація	ModelSim + OSVVM + UVVM	Оцінки компонент власного вектора	Нормалізовані оцінки вектора пріоритету
ModelSim	1	1/2	3	3	1/4	1,023836	0,145561
QuestaSim	2	1	4	4	1/3	1,605483	0,228255
Quartus	1/3	1/4	1	1	1/7	0,412234	0,058608
Формальна верифікація	1/3	1/4	1	1	1/7	0,412234	0,058608
ModelSim + OSVVM + UVVM	4	3	7	7	1	3,579938	0,508967
Сума	7,6667	5,0000	16,0000	16,0000	1,8690	7,033726	

Відношення узгодженості (ВУ) =	**1,87%**	Повинно бути < 10%, допускається < 20%.

Функціональність

ПО Верифікації	К-сть функцій
ModelSim	6
QuestaSim	5
Quartus	10
Формальна верифікація	12
ModelSim + OSVVM + UVVM	8

	ModelSim	QuestaSim	Quartus	Формальна верифікація	ModelSim + OSVVM + UVVM	Оцінки компонент власного вектора	Нормалізовані оцінки вектора пріоритету
ModelSim	1	2	1/5	1/9	1/5	0,388839	0,053632
QuestaSim	1/2	1	1/9	1/8	1/7	0,250789	0,034591
Quartus	5	9	1	2	1/2	2,141127	0,295324
Формальна верифікація	9	8	1/2	1	1/2	1,782602	0,245873
ModelSim + OSVVM + UVVM	5	7	2	2	1	2,686740	0,370580
Сума	20,5000	27,0000	3,8111	5,2361	2,3429	7,250097	

Відношення узгодженості (ВУ) =	**7,02%**	Повинно бути < 10%, допускається < 20%.

Вартість для користувача

ПО Верифікації	Умовні одиниці
ModelSim	7
QuestaSim	6
Quartus	8
Формальна верифікація	2
ModelSim + OSVVM + UVVM	5

	ModelSim	QuestaSim	Quartus	Формальна верифікація	ModelSim + OSVVM + UVVM	Оцінки компонент власного вектора	Нормалізовані оцінки вектора пріоритету
ModelSim	1	4	8	9	2	3,565205	0,455031
QuestaSim	1/4	1	5	4	1/4	1,045640	0,133456
Quartus	1/8	1/5	1	1/2	1/8	0,274640	0,035053
Формальна верифікація	1/9	1/4	2	1	1/7	0,380125	0,048516
ModelSim + OSVVM + UVVM	1/2	4	8	7	1	2,569470	0,327944
Сума	1,9861	9,4500	24,0000	21,5000	3,5179	7,835079	

Відношення узгодженості (ВУ) =	**4,53%**	Повинно бути < 10%, допускається < 20%.